How to get the most out of WAS 8.5.5

Turning the new capabilities into real value for you and your company

WebSphere User Group
September 2013

Alan Chambers
WebSphere Consultant

Topics

- A short review of what was new in V8.5.5
- Introduction to Intelligent Management
- How to get started with
  - Autonomic Performance Management
  - Health Management
  - Application Edition Management
- Summary

Who am I?

- Former IBMer
- Founder of the UK WUG
  - and currently Acting Chairman
- Well known for WebSphere foundation technologies
  - WAS, WVE and WXS
- Now a freelancer working with these and other companies:

ORBITAL

IBM Partner Marketing Platinum Award 2012

W3 Partnership

Expertise to Inspire Confidence

ARROW ECS EDUCATION

DIEGESIS

Verhoef TRAINING

What's new in 8.5.5?

What is V8.5.5?

- Is it a new release - or just a bunch of fixes?

- Answer:
  - A bit of both
  - It *is* the next Fix Pack after V8.5.0.2
  - But there's significant new function too

    ...and a new licence option

So what's actually new?

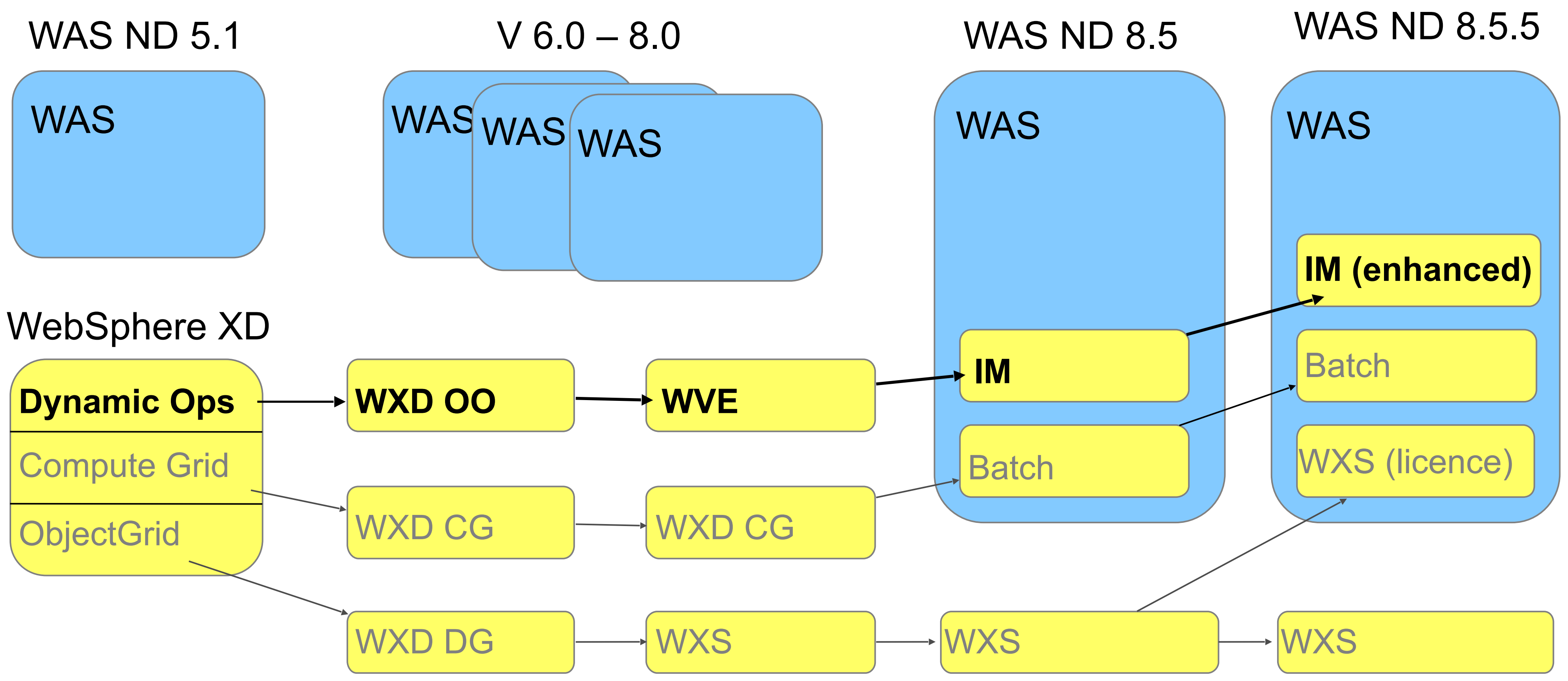
- Liberty enhancements (many of these)
  - Including EJB 3.1 Lite, JMS and WMQ, Dynacache, WS enhancements etc., etc.
  - New licence: *Liberty Core*
- Intelligent Management support in HTTP Plug-in
  - Removes need for On-Demand Router in many IM topologies
  - Use this to benefit from dynamic application placement, health management and application edition management...

    ...with no change to your topology!

Intelligent Management:

Introduction and history 2004 - 2013

WAS Intelligent Management isn't new – it just keeps changing name!



What does Intelligent Management add?

Performance Management

Health Management

Application Edition Management

Various minor WAS enhancements

What does IM do for you? (1) Performance Management

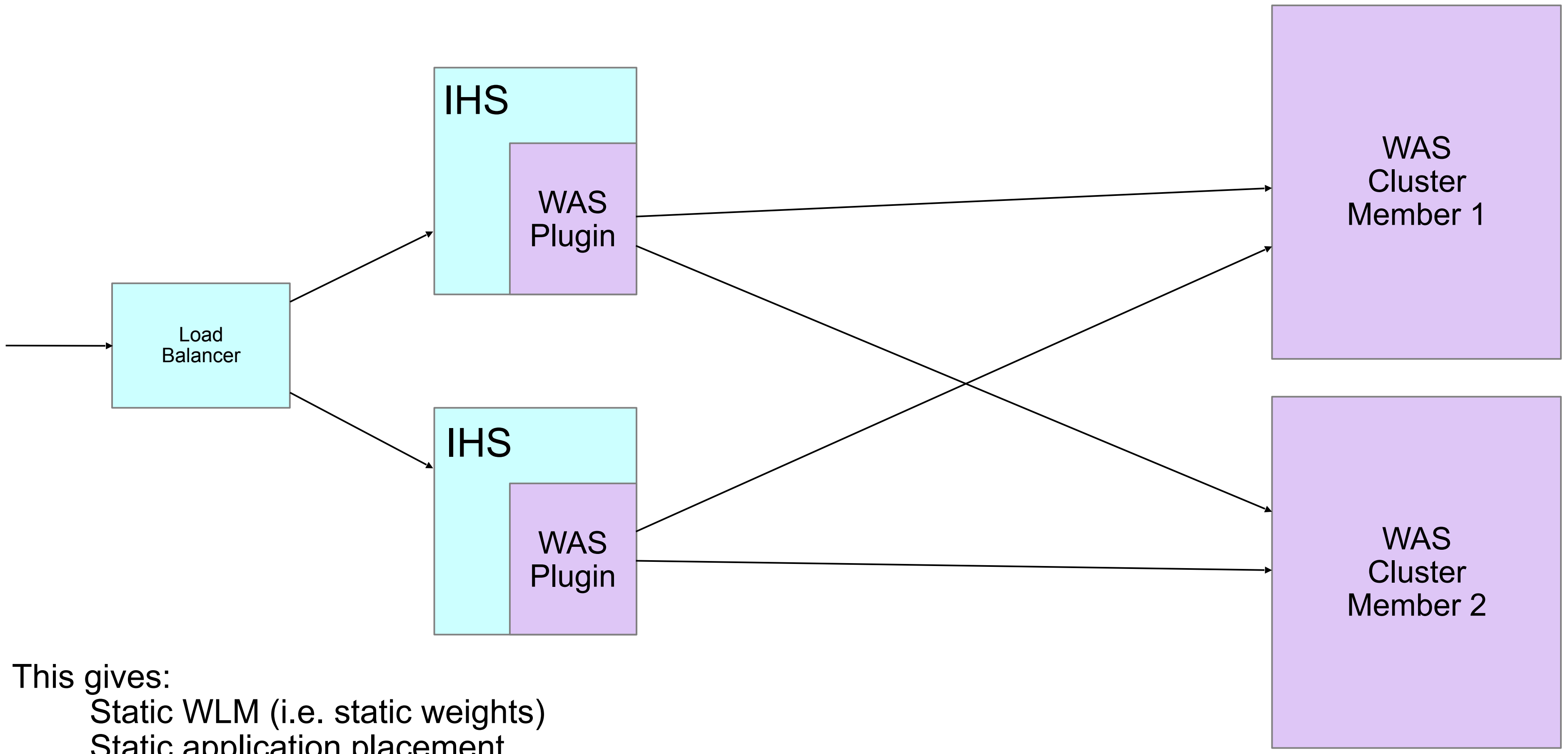
- You define response time goals & priorities
  - Per application, part of application, user group, etc..
  - “Service Policy”
- You pool some or all of your servers - IM then:
  - Dynamically allocates servers to applications based on current load, response times vs. goals, priorities (if resources constrained)
  - Controls request flow and workload manages them across whichever servers are currently running the application

What does IM do for you? (1) Performance Management

- WAS's policy-based performance management gives you:
  - Ability to run servers at higher utilisations than with other approaches
    - Saving you the cost of hardware, software licences, maintenance, operations, power, cooling, floor space…
  - Provide more predictable service to users
    - Against agreed response time objectives
  - The ability to record data on usage
    - so that application owners can be charged for their actual use of the shared infrastructure

Where does this happen?

Traditional WAS clustering

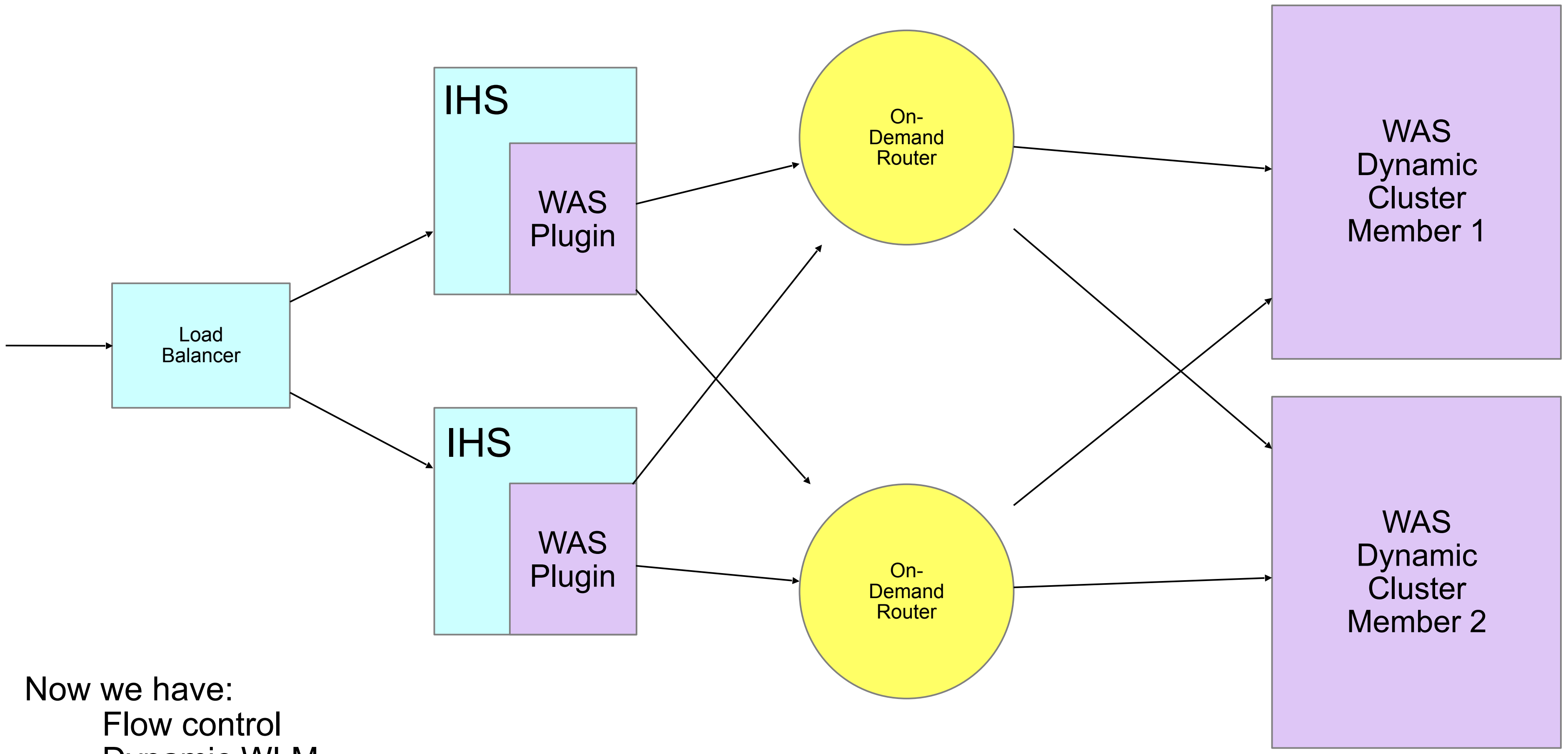


This gives:

- Static WLM (i.e. static weights)
- Static application placement

Where does this happen?

Traditional XD/WVE/WAS dynamic clustering (pre 8.5.5)

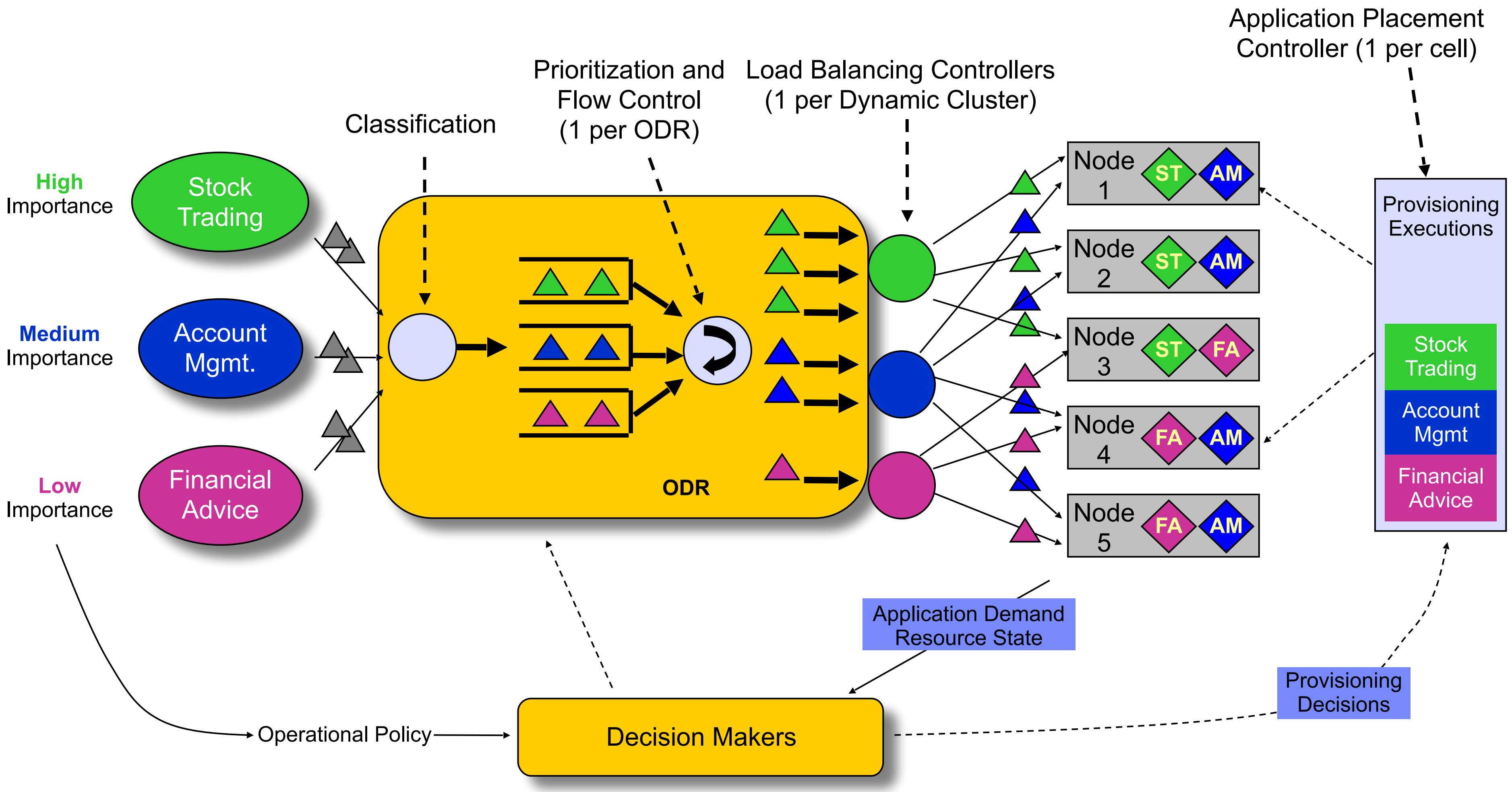


Now we have:
- Flow control
- Dynamic WLM
- Goal-based response time management
- Dynamic application placement

The On-Demand Router

- An HTTP proxy that:
  - Monitors response times against goals
  - Prioritises incoming requests to manage response times from second to second
  - Adjusts WLM weights based on real-time data about response times and CPU load on target servers
  - Manages request flow to avoid overloading
  - Usually hosts the *Autonomic Request Flow Manager* service (ARFM)
  - ARFM requests extra resources from *Application Placement Manager* service when needed to achieve goals for an application
- You need >=2 to avoid a single point of failure
  - Operating in an active/active mode

How it all works



What does IM do for you? (2) Health Management

- What is Health Management?
  - Proactively deal with application and application infrastructure issues before they become acute problems … automatically
  - Health conditions and associated corrective actions
    - Conditions: e.g. memory leak, excessive response time, storm drain, custom conditions
    - Actions: e.g. take heap dump, put server into maintenance mode, restart server, etc..
  - Requires application and infrastructure insight!
- Results in:
  - Better availability
  - Less administration required
  - Satisfied end users
  - Continuous service, even with “broken” applications

Health Management in action

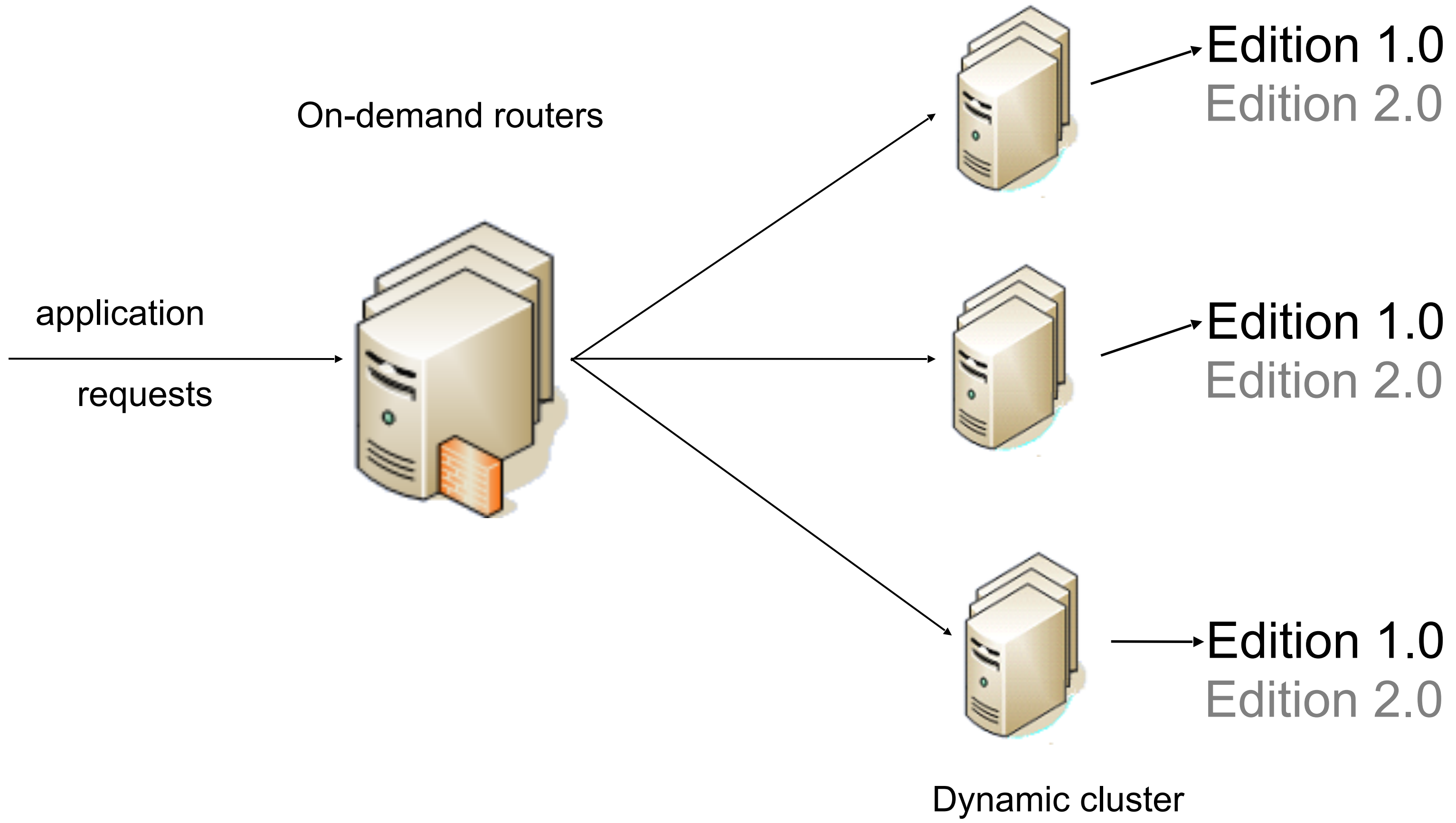
- What happens when a health condition is detected?
  - Affected server usually taken out of service
    - Either in maintenance mode, or while it restarts
  - Application Placement Controller (APC) starts new instance of application (i.e. DC member) to make up for lost capacity
    - ODRs route requests only to currently active instances, so no loss of service to users
  - Once affected server has restarted, ODR routes requests to it again

- Bottom line: no loss of service to users

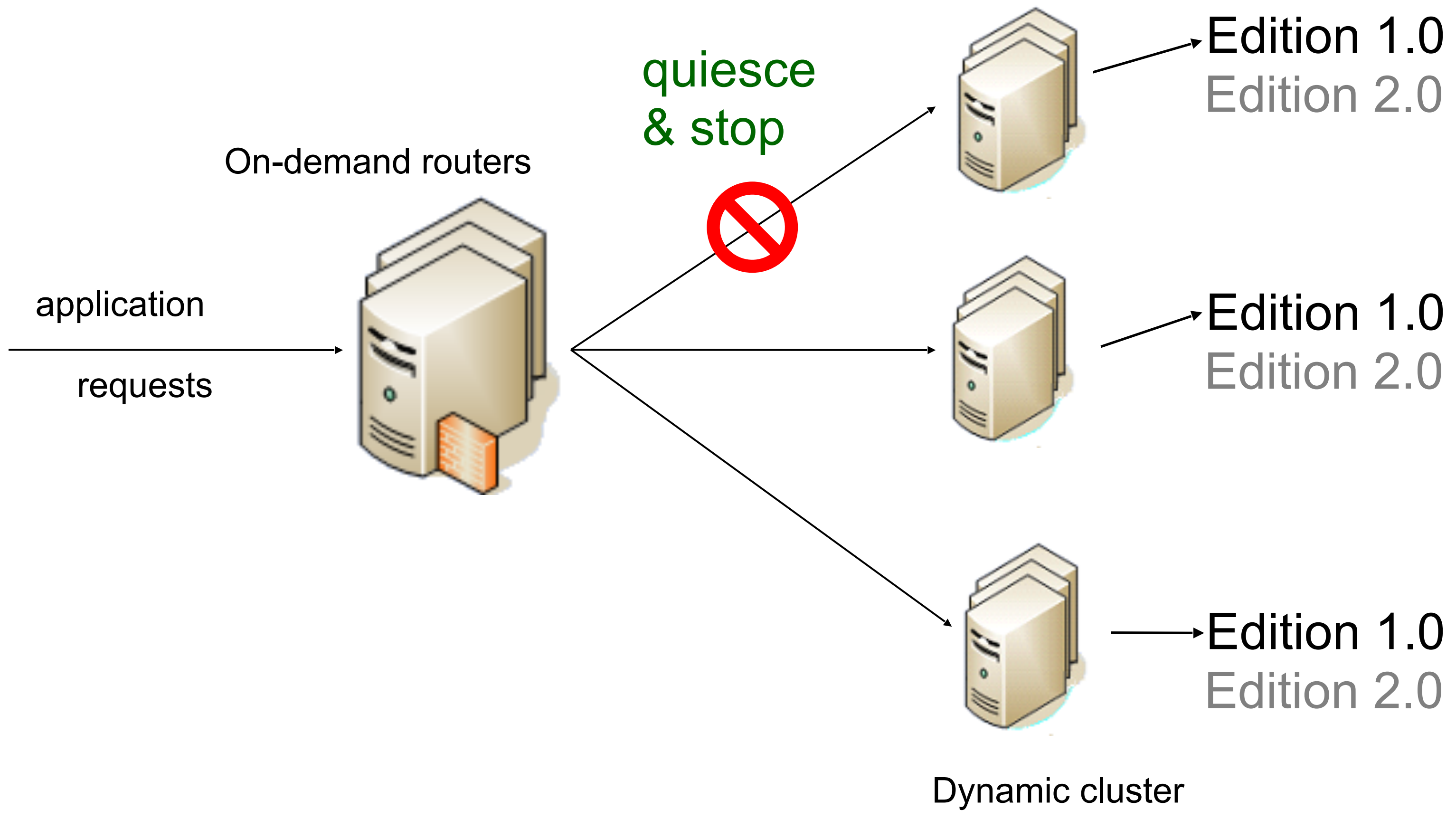
What does IM do for you? (2) Application Edition Management

- An edition is a distinct instance of a JEE 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

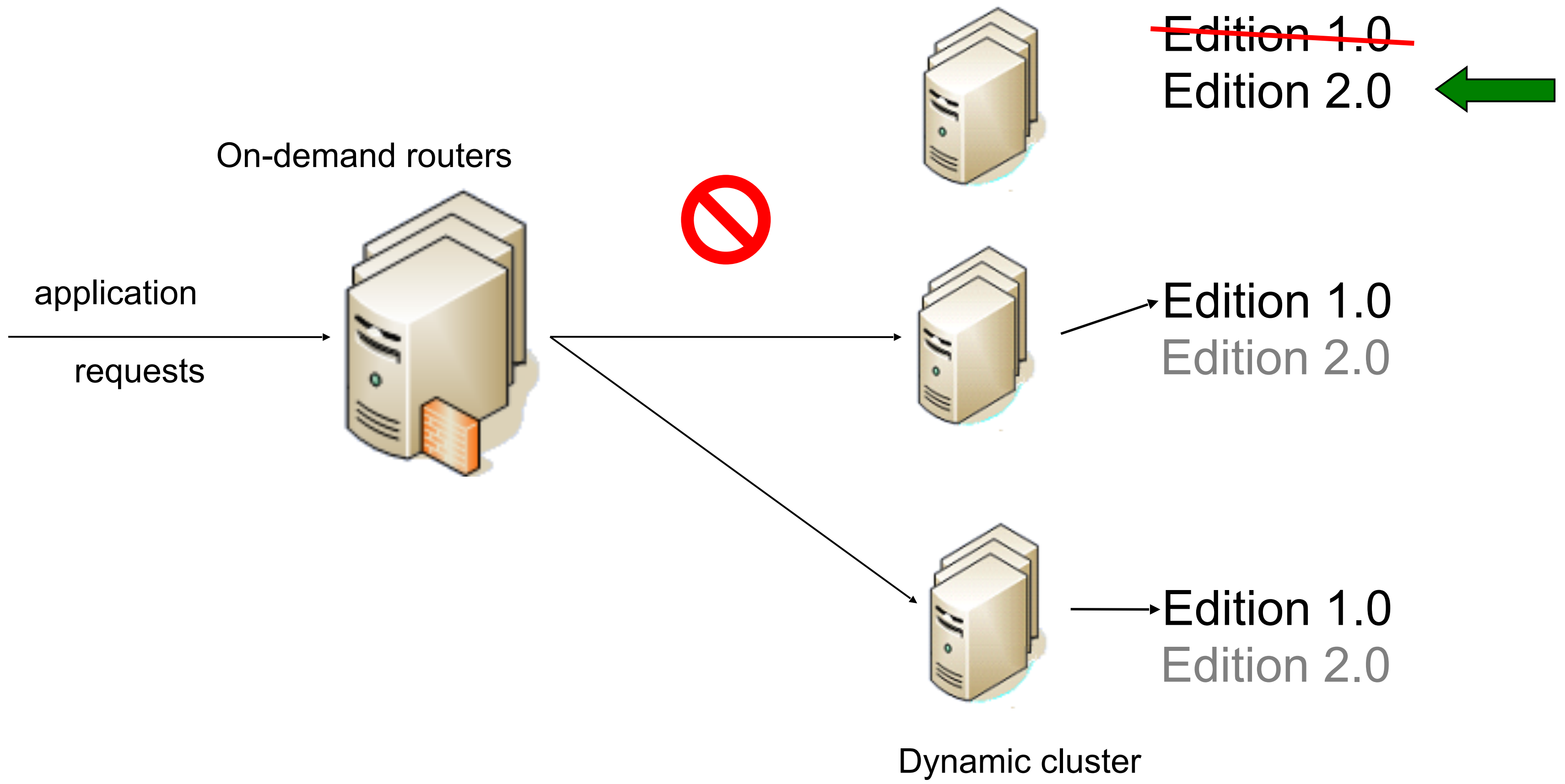
Operational Scenarios – Grouped Rollout



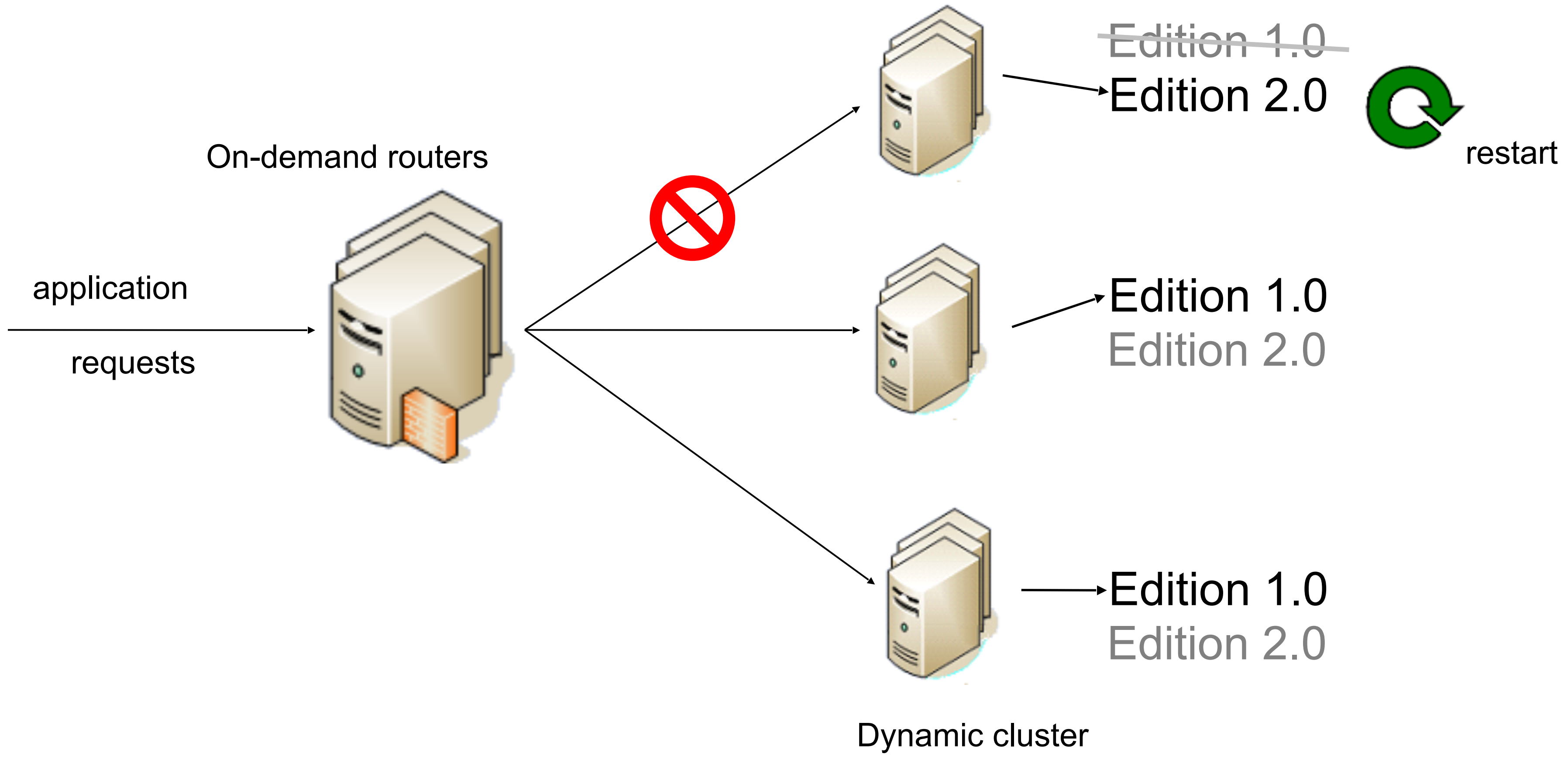
Operational Scenarios – Grouped Rollout



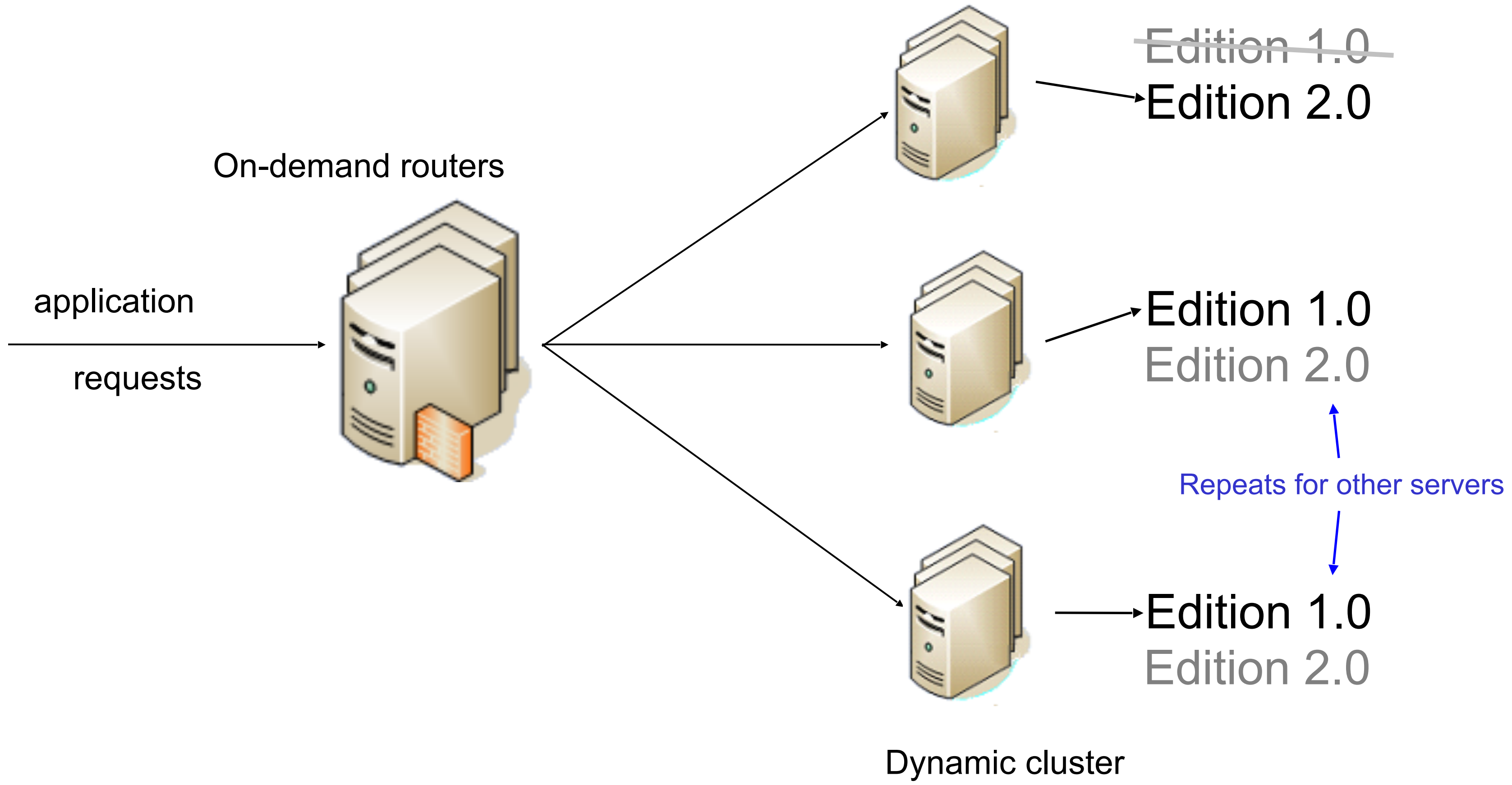
Operational Scenarios: Grouped Rollout



Operational Scenarios – Grouped Rollout



Operational Scenarios – Grouped Rollout



Validation Mode

- Creates a ‘clone’ of the deployment target (Dynamic Cluster) in production environment and deploys new version
- Use routing policies to control edition visibility
  - Restrict new edition to only certain users, for example
- Use rollout function to move edition from validation mode to production
  - Edition deactivated on ‘clone’ environment
  - Edition rolled out on original deployment targets
- Multiple applications can share the same validation cluster
  - These applications must be different and cannot be different editions of the same application.

So - it's not just about "Resiliency"

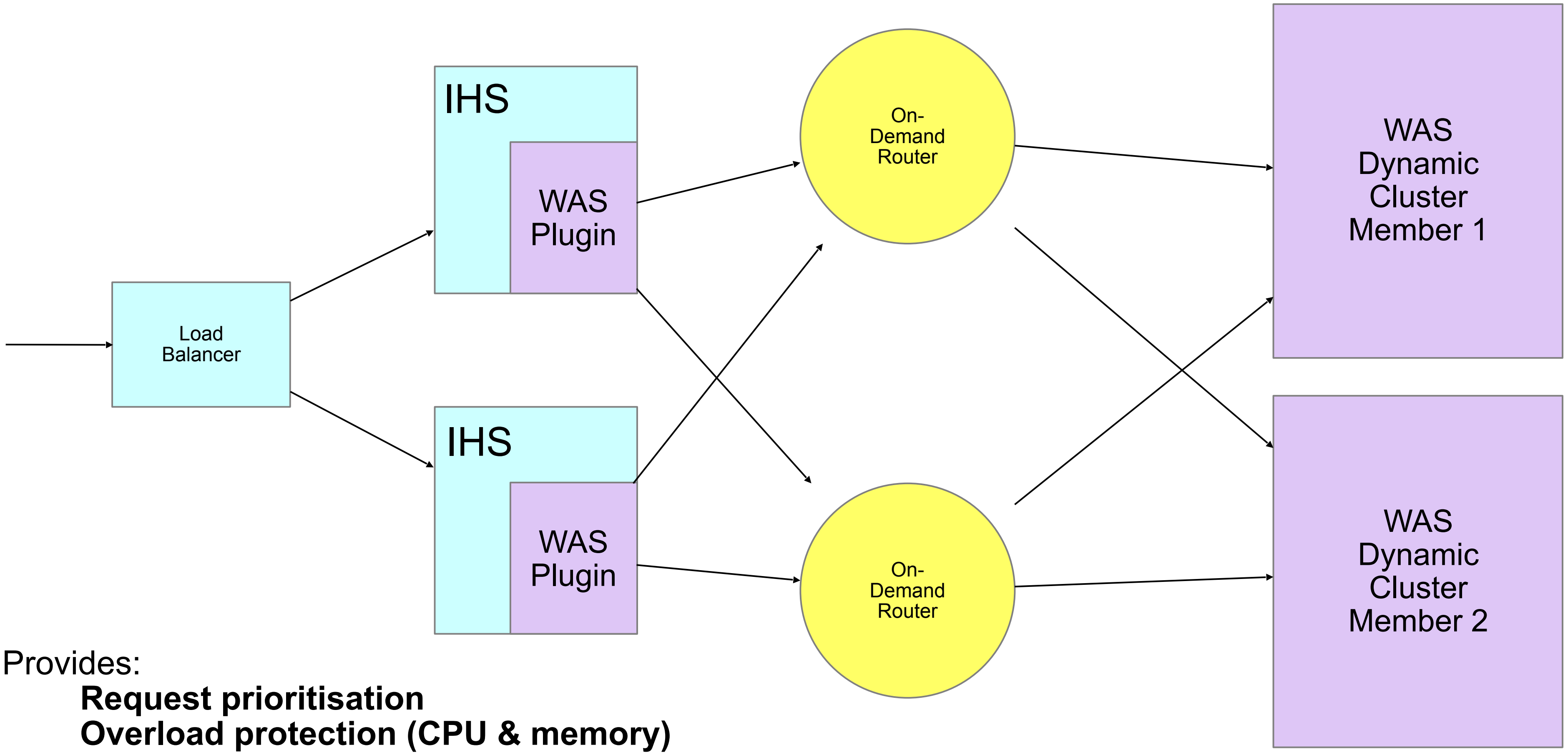
Intelligent Management gives you:

- Performance Management
  - Save money by running servers at higher utilisation
  - Make adding and removing capacity easier and less finger-in-the-air
- Health Management
  - Provide more resilient service to users
  - Reduce support/operations costs
- Application Edition Management
  - More productive application validation and upgrades

Why is WAS 8.5.5 so interesting?

Traditional WVE/WAS 8.5 IM Topology

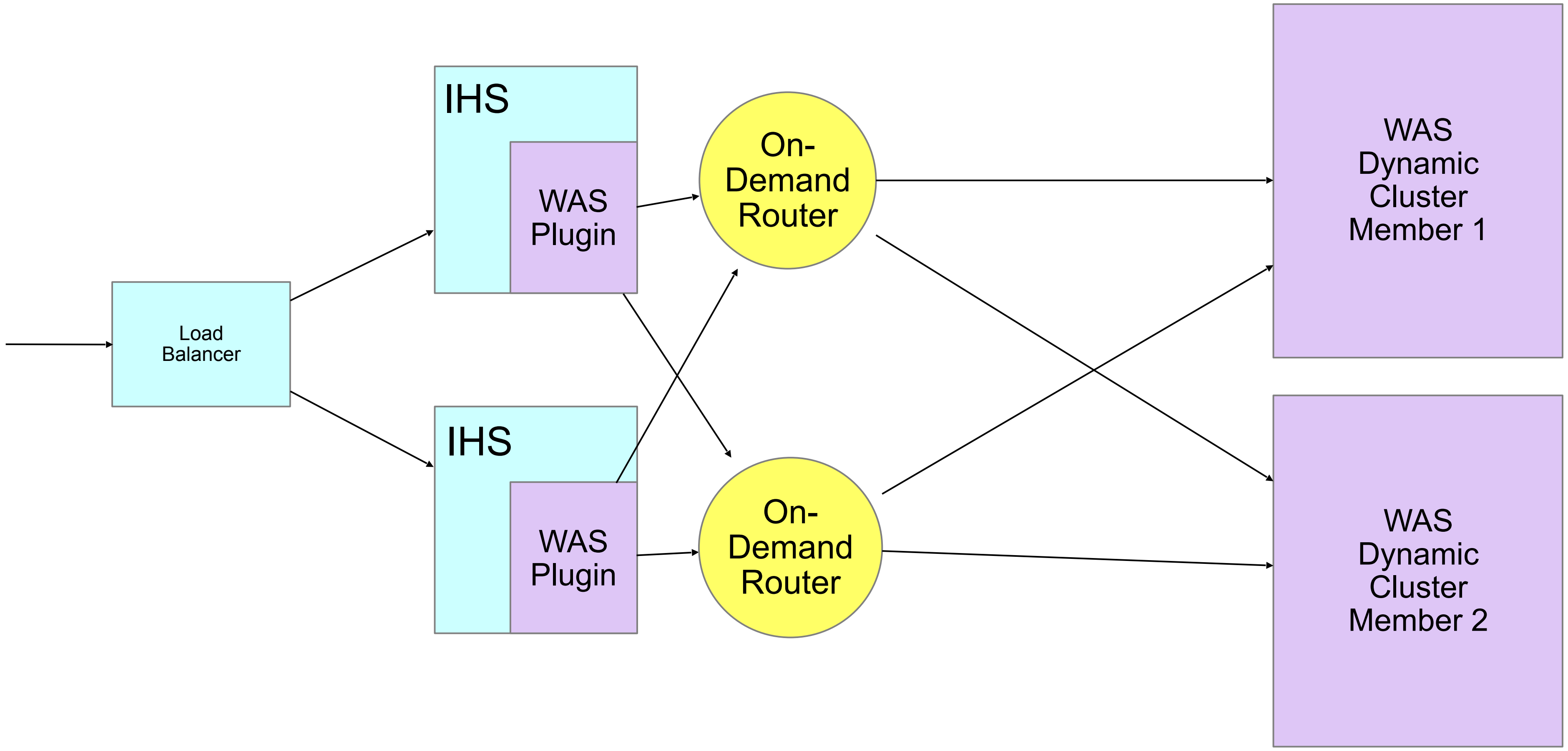
Requires ODR tier – h/w and licences



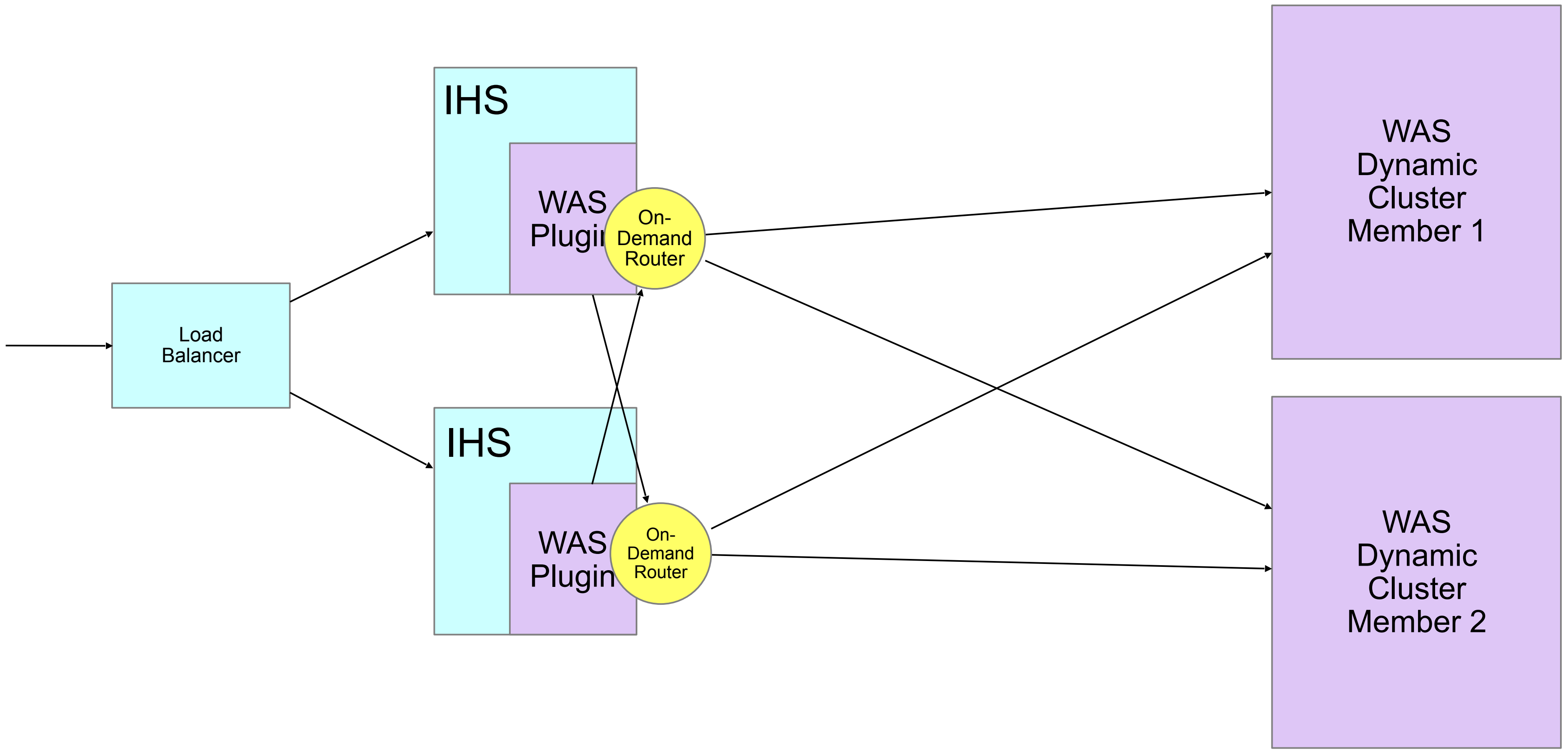
Provides:

- **Request prioritisation**
- **Overload protection (CPU & memory)**
- **Dynamic WLM**
- **Goal-based response time management**
- **Dynamic application placement**

Traditional WVE/WAS 8.5 IM Topology

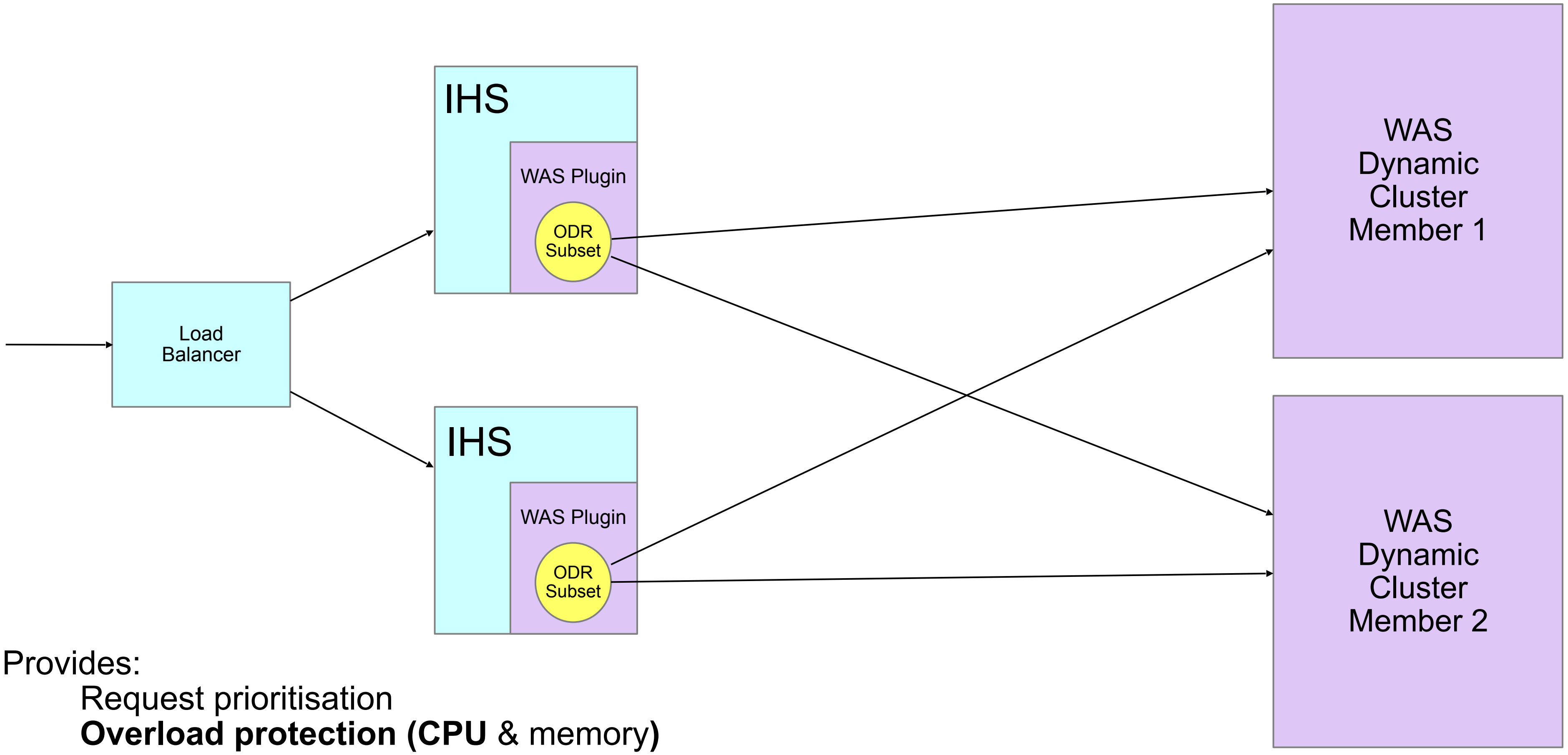


Traditional WVE/WAS 8.5 IM Topology



Simplified topology with WAS 8.5.5

No ODRs needed, so no extra servers and licences



Provides:

- Request prioritisation
- **Overload protection (CPU** & memory**)**
- **Dynamic WLM**
- **Goal-based response time management**
- **Dynamic application placement**

How to implement Intelligent Management on your existing configuration

Implementing this: first steps

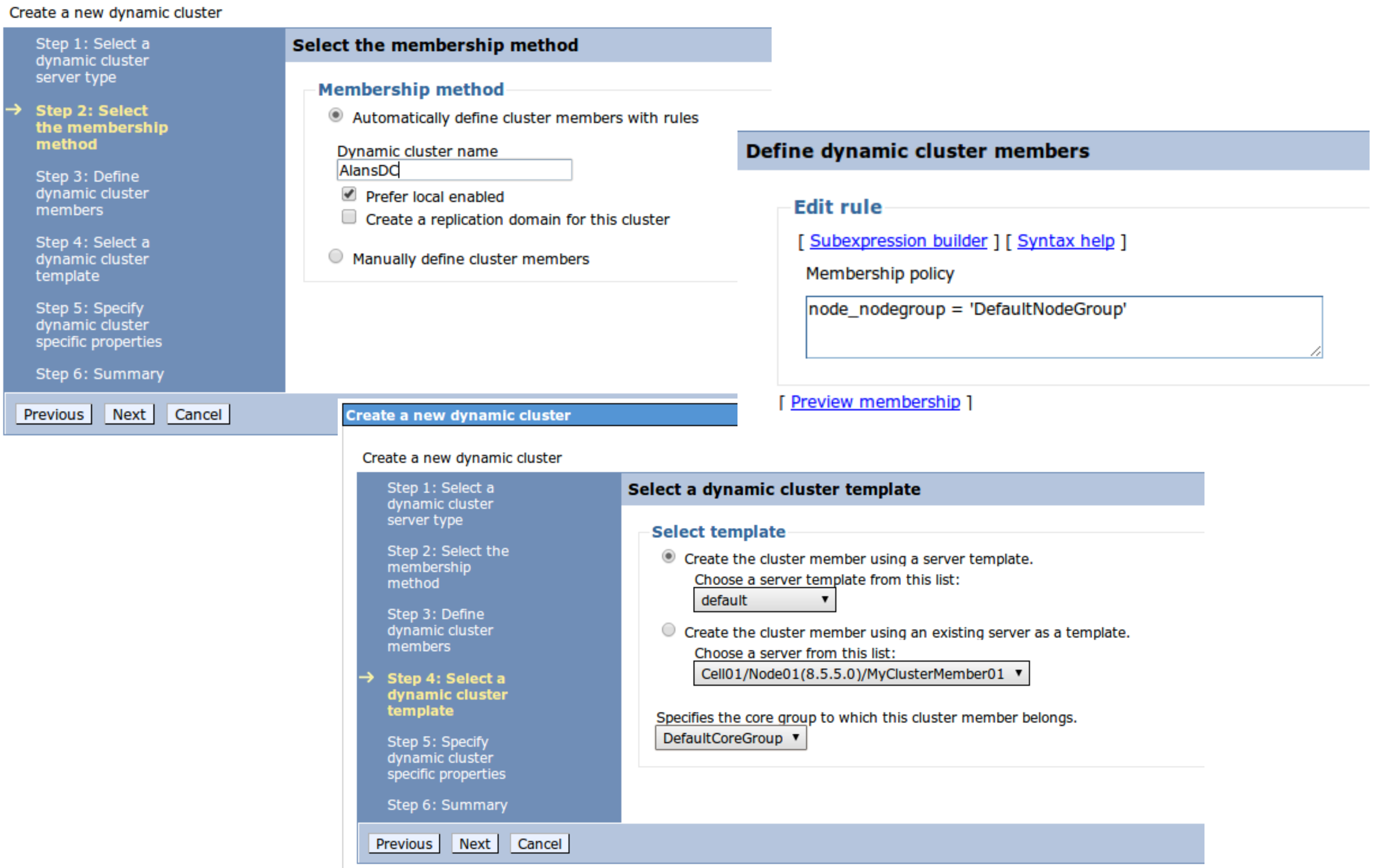
1. Upgrade all servers in the cell to V8.5.5
    - From 8.5 this is like a Fix Pack install
2. Define a Dynamic Cluster for one application
    - including as many of your nodes as possible (preferably all)
    - Set this DC to *Automatic* or *Supervised* - now or later
3. Map the application to this DC instead of the current static cluster
4. Enable Intelligent Management on your web server plug-ins
5. Configure Health Policies or Service Policies as desired
    - see below for details!

Creating a Dynamic Cluster

- What is a Dynamic Cluster (DC)?
  - Just like a Static Cluster:
    - It's a collection of server defined on several nodes
    - Applications mapped to a DC are deployed to all the servers within it
    - WLM and failover occur across the members
  - *Unlike* a Static Cluster:
    - You can't start or stop a DC – instead individual members are started and stopped as required (on-demand) by the Application Placement Controller
    - Rules determine which nodes it incorporates, vertical stacking level, application isolation...
    - Best practice is to map only one application to each DC

Tip! ***DCs make it much easier to keep your cluster configurations consistent***

Defining a Dynamic Cluster



Create a new dynamic cluster

Step 1: Select a dynamic cluster server type
→ Step 2: Select the membership method
Step 3: Define dynamic cluster members
Step 4: Select a dynamic cluster template
Step 5: Specify dynamic cluster specific properties
Step 6: Summary
Previous
Next
Cancel

Select the membership method

Membership method

(●) Automatically define cluster members with rules

Dynamic cluster name: AlansDC

[✓] Prefer local enabled

[ ] Create a replication domain for this cluster

( ) Manually define cluster members

Define dynamic cluster members

Edit rule

[Subexpression builder] [Syntax help]

Membership policy: node_nodegroup = 'DefaultNodeGroup'

Create a new dynamic cluster

Create a new dynamic cluster

Step 1: Select a dynamic cluster server type
Step 2: Select the membership method
Step 3: Define dynamic cluster members
→ Step 4: Select a dynamic cluster template
Step 5: Specify dynamic cluster specific properties
Step 6: Summary
Previous
Next
Cancel

Select a dynamic cluster template

Select template

(●) Create the cluster member using a server template.

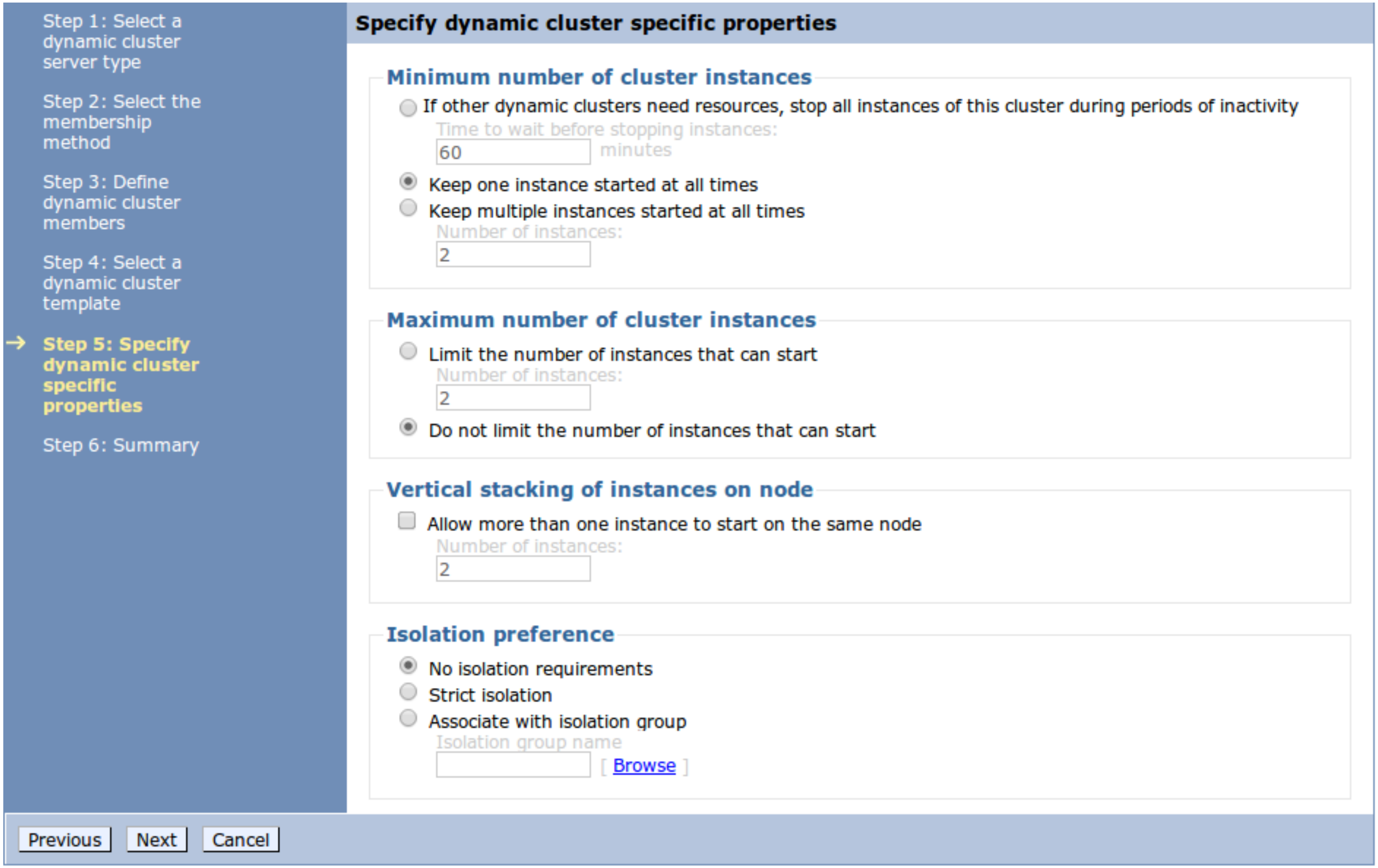
Choose a server template from this list: default

( ) Create the cluster member using an existing server as a template.

Choose a server from this list: Cell01/Node01(8.5.5.0)/MyClusterMember01

Specifies the core group to which this cluster member belongs. DefaultCoreGroup

Dynamic cluster properties



Step 1: Select a dynamic cluster server type

Step 2: Select the membership method

Step 3: Define dynamic cluster members

Step 4: Select a dynamic cluster template

→ **Step 5: Specify dynamic cluster specific properties**

Step 6: Summary

Specify dynamic cluster specific properties

Minimum number of cluster instances

( ) If other dynamic clusters need resources, stop all instances of this cluster during periods of inactivity

Time to wait before stopping instances: 60 minutes

[✓ Keep one instance started at all times]

( ) Keep multiple instances started at all times

Number of instances: 2

Maximum number of cluster instances

( ) Limit the number of instances that can start

Number of instances: 2

[✓ Do not limit the number of instances that can start]

Vertical stacking of instances on node

☐ Allow more than one instance to start on the same node

Number of instances: 2

Isolation preference

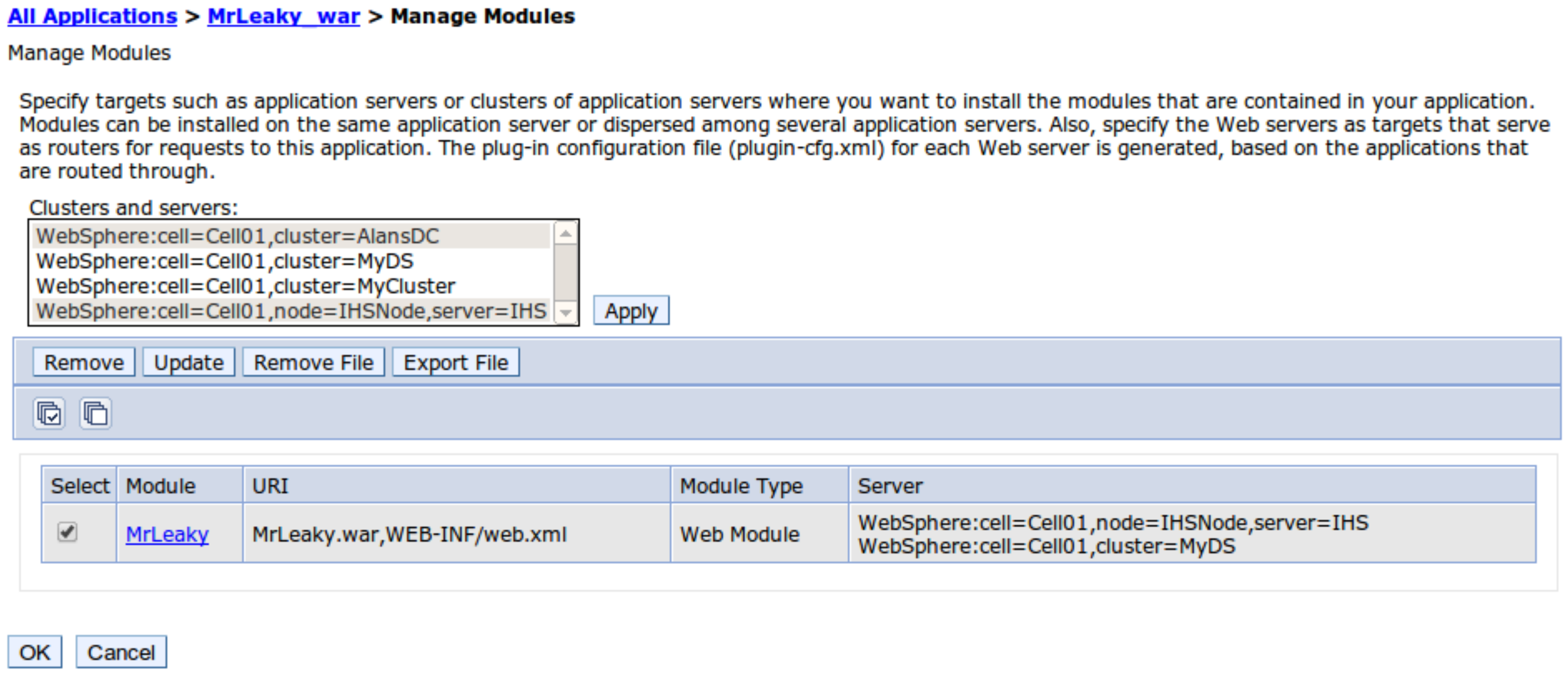
[✓ No isolation requirements]

( ) Strict isolation

( ) Associate with isolation group

Isolation group name

Map the application to your DC



All Applications > MrLeaky_war > Manage Modules

Manage Modules

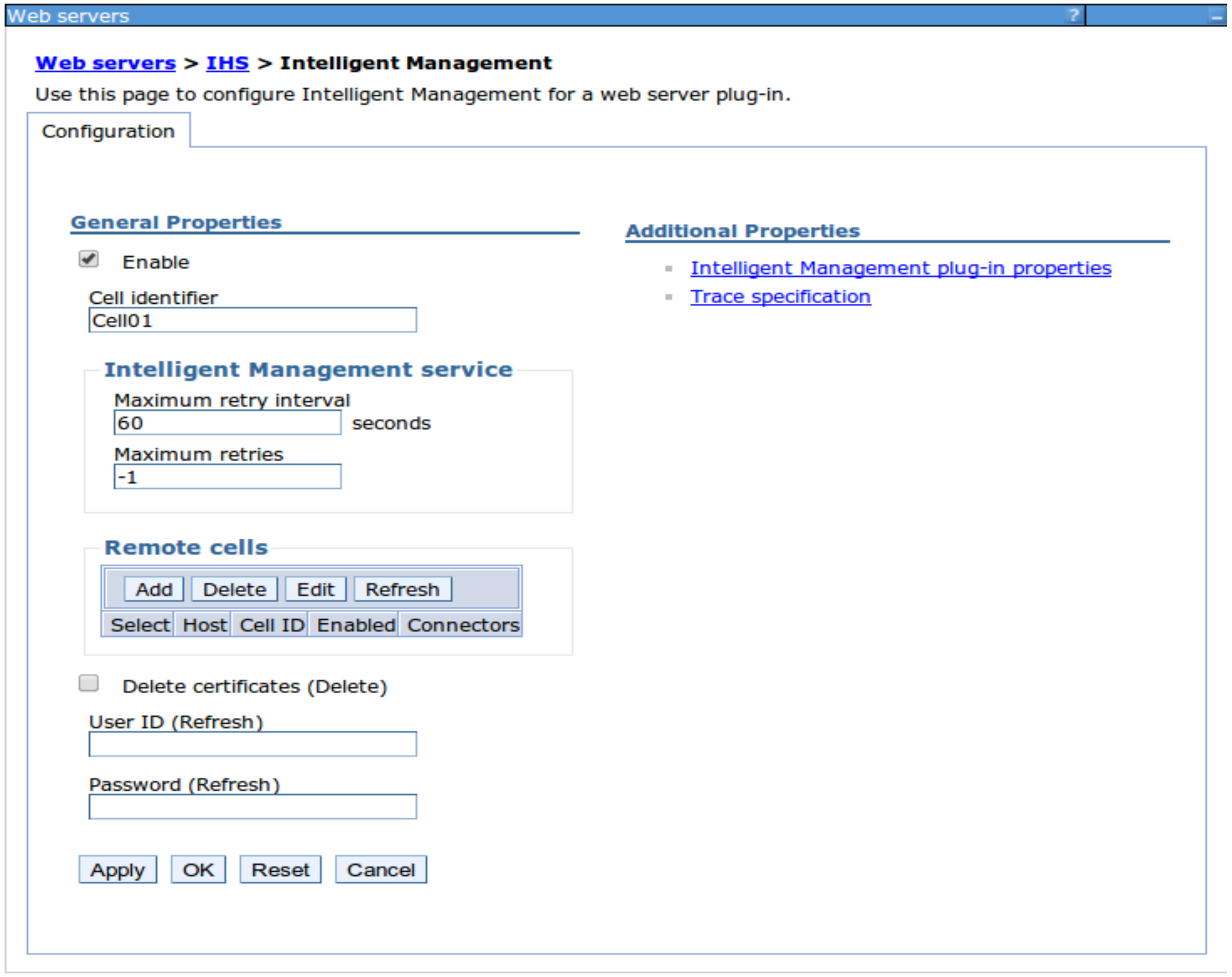
Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

```
WebSphere:cell=Cell01,cluster=AlansDC
WebSphere:cell=Cell01,cluster=MyDS
WebSphere:cell=Cell01,cluster=MyCluster
WebSphere:cell=Cell01,node=IHSNode,server=IHS
```

Select	Module	URI	Module Type	Server
✓	MrLeaky	MrLeaky.war,WEB-INF/web.xml	Web Module	WebSphere:cell=Cell01,node=IHSNode,server=IHS WebSphere:cell=Cell01,cluster=MyDS

Enabling IM on the Web Server plug-in



Web servers

Web servers > IHS > **Intelligent Management**

Use this page to configure Intelligent Management for a web server plug-in.

Configuration

General Properties

[✓] Enable

Cell identifier: Cell01

Intelligent Management service

Maximum retry interval: 60 seconds

Maximum retries: -1

Remote cells

Select	Host	Cell ID	Enabled	Connectors

[ ] Delete certificates (Delete)

User ID (Refresh):

Password (Refresh):

Additional Properties

- Intelligent Management plug-in properties
- Trace specification

Configuring performance management for an application

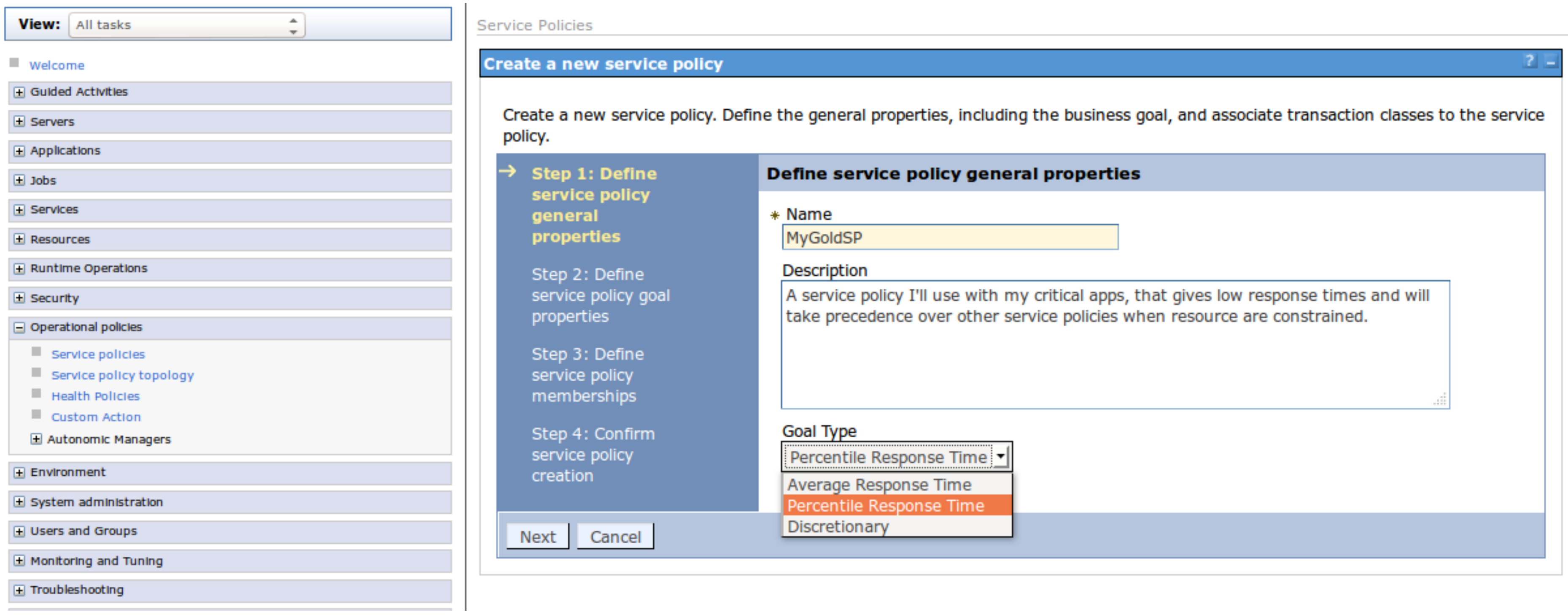
or

How to improve service levels and save money!

Configuring WAS for Performance Management

- The story so far...
  - We have a “pool” of servers in the form of a Dynamic Cluster defined over many nodes
  - WAS can start and stop these as desired to allocated more or less resource to the application(s) deployed on them
    - To adjust capacity and/or performance
- But...
  - WAS doesn't know what we want it to aim for
    - So how can it decide how much resource to allocate to different application?
- This is what Service Policies are for:
  - They define response time goals and priorities
  - They are associated with applications
    - Or parts of applications (example: “buying” vs. “browsing”)
    - Or groups of users (“Gold” customers vs. the rest,)

Defining a Service Policy (1)



Service Policies

Create a new service policy

Create a new service policy. Define the general properties, including the business goal, and associate transaction classes to the service policy.

Step 1: Define service policy general properties
Step 2: Define service policy goal properties
Step 3: Define service policy memberships
Step 4: Confirm service policy creation

Define service policy general properties

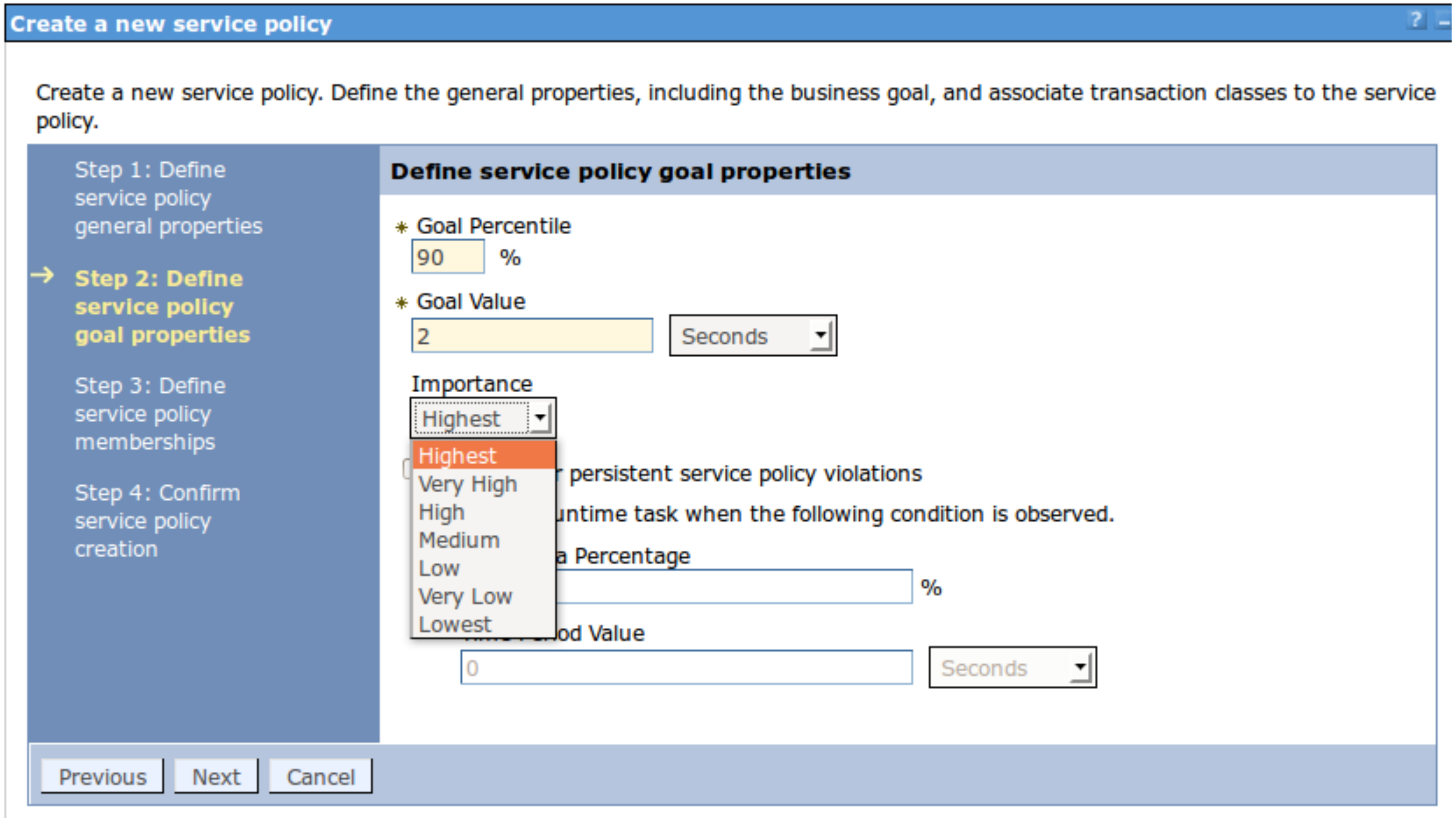
* Name: MyGoldSP

Description: A service policy I'll use with my critical apps, that gives low response times and will take precedence over other service policies when resource are constrained.

Goal Type: Percentile Response Time

- Average response time: what it says on the tin
- Percentile response time: e.g. 2 secs, 90% of the time
- Discretionary: best efforts after all other policies satisfied

Defining a Service Policy (2)


Create a new service policy

Create a new service policy. Define the general properties, including the business goal, and associate transaction classes to the service policy.

Step 1: Define service policy general properties

→ **Step 2: Define service policy goal properties**

Step 3: Define service policy memberships

Step 4: Confirm service policy creation

Define service policy goal properties

* Goal Percentile: 90 %

* Goal Value: 2 Seconds

Importance: Highest

Highest
Very High
High
Medium
Low
Very Low
Lowest

r persistent service policy violations

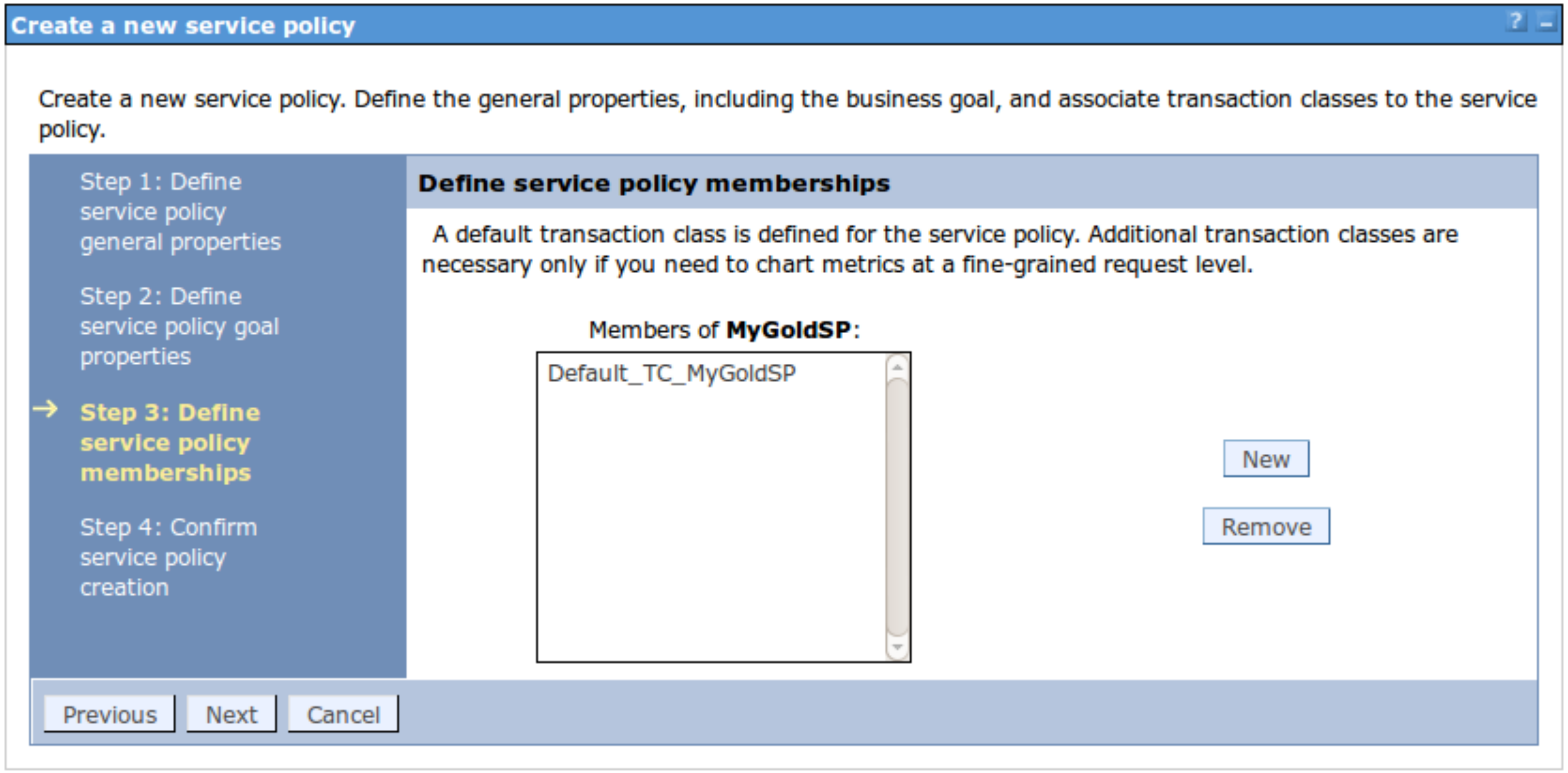
ıntime task when the following condition is observed.

a Percentage: %

od Value: 0 Seconds

Importance helps IM decide how to act when resources prevent all SPs from being satisfied

Defining a Service Policy (3)



Create a new service policy

Create a new service policy. Define the general properties, including the business goal, and associate transaction classes to the service policy.

Step 1: Define service policy general properties

Step 2: Define service policy goal properties

→ **Step 3: Define service policy memberships**

Step 4: Confirm service policy creation

Define service policy memberships

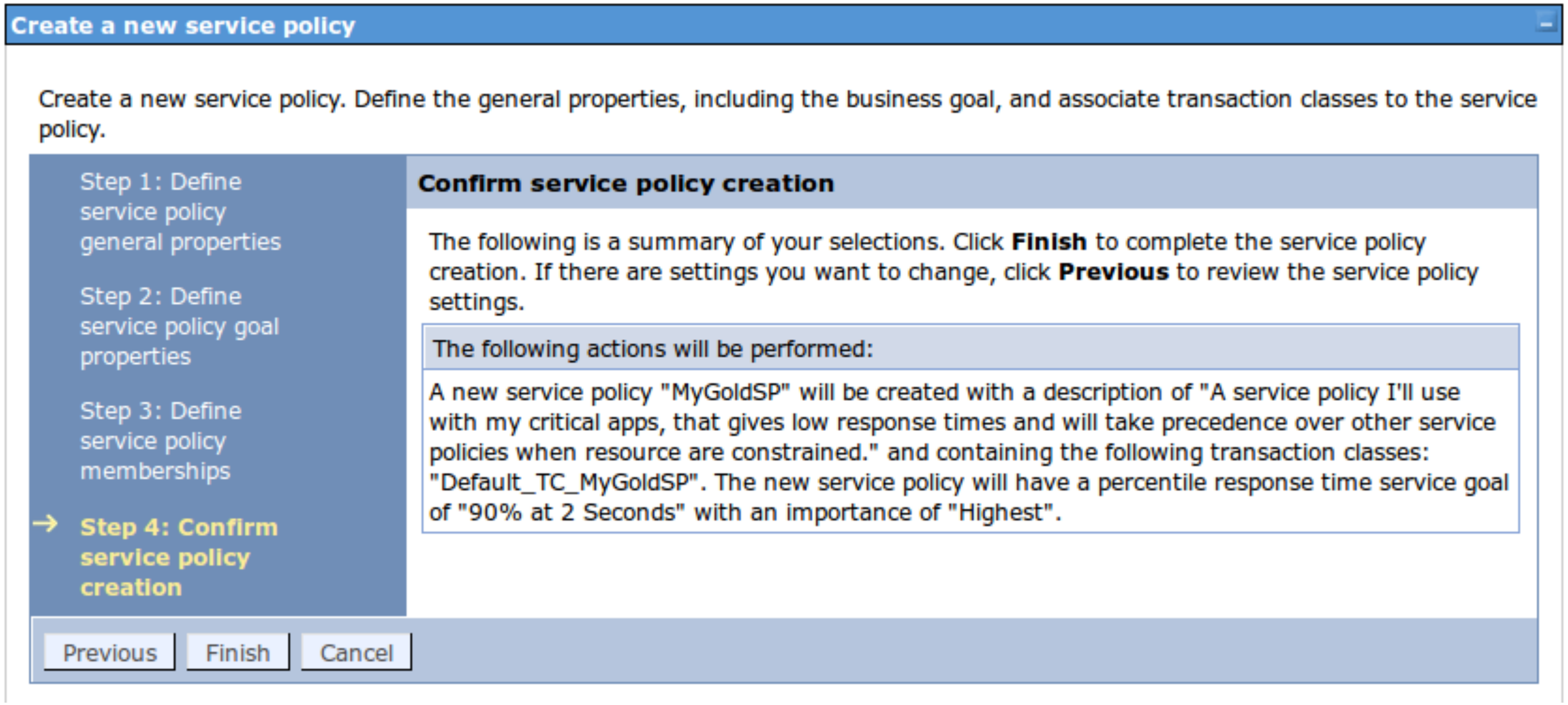
A default transaction class is defined for the service policy. Additional transaction classes are necessary only if you need to chart metrics at a fine-grained request level.

Members of **MyGoldSP**:

Default_TC_MyGoldSP

- Accept the default *Transaction Class* here
  - not many people understand the point of these, so why should you!

Defining a Service Policy (4)


Create a new service policy

Create a new service policy. Define the general properties, including the business goal, and associate transaction classes to the service policy.

Step 1: Define service policy general properties
Step 2: Define service policy goal properties
Step 3: Define service policy memberships
→ Step 4: Confirm service policy creation

Confirm service policy creation

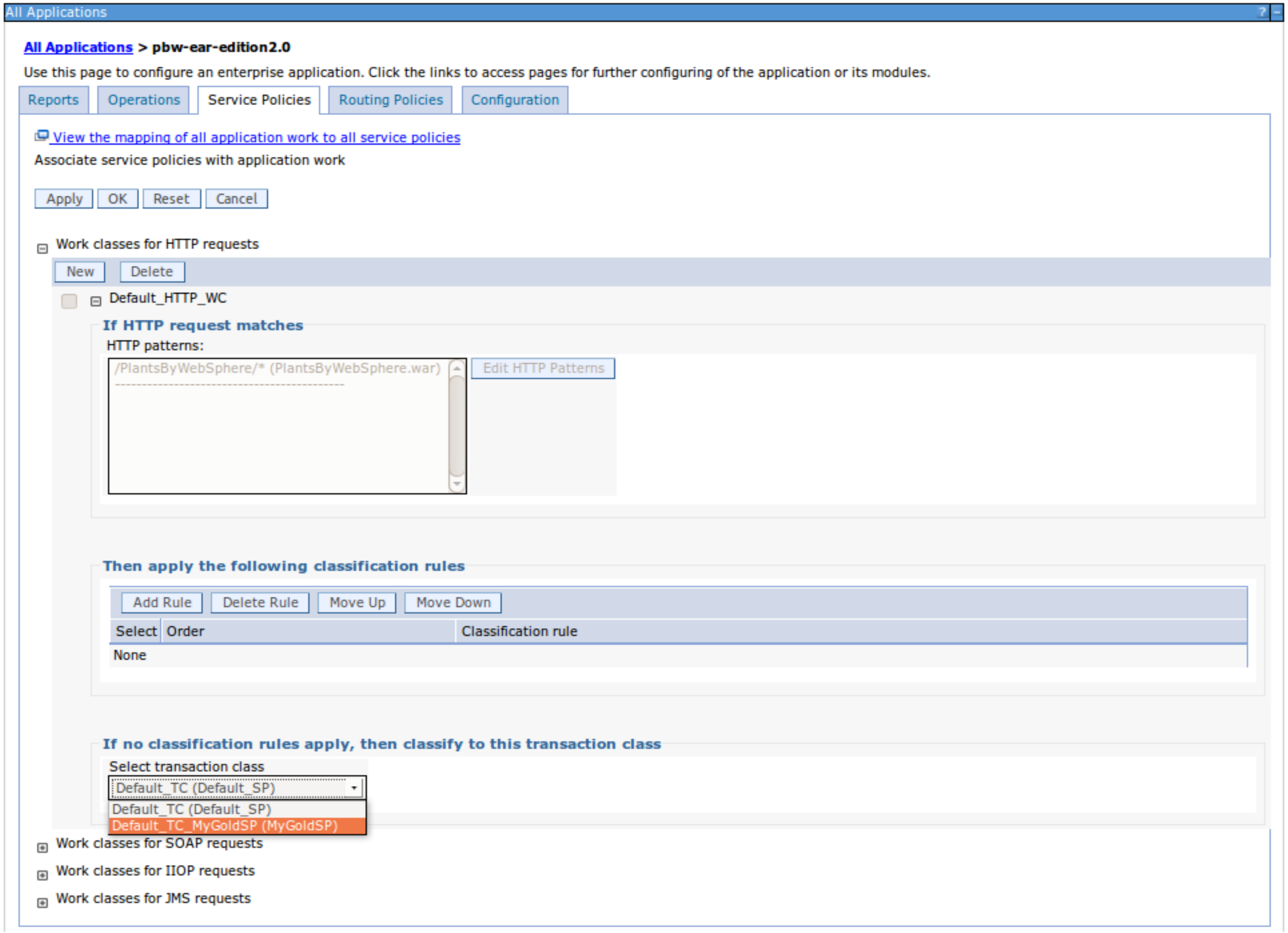
The following is a summary of your selections. Click **Finish** to complete the service policy creation. If there are settings you want to change, click **Previous** to review the service policy settings.

The following actions will be performed:

A new service policy "MyGoldSP" will be created with a description of "A service policy I'll use with my critical apps, that gives low response times and will take precedence over other service policies when resource are constrained." and containing the following transaction classes: "Default_TC_MyGoldSP". The new service policy will have a percentile response time service goal of "90% at 2 Seconds" with an importance of "Highest".

- That's it!
  - Click *Finish* and we have an SP we can use for as many applications as we like

Applying an SP to an application



All Applications > pbw-ear-edition2.0

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

View the mapping of all application work to all service policies

Associate service policies with application work

Work classes for HTTP requests

Default_HTTP_WC

If HTTP request matches

HTTP patterns:

/PlantsByWebSphere/* (PlantsByWebSphere.war)

Then apply the following classification rules

Select	Order	Classification rule
None		

If no classification rules apply, then classify to this transaction class

Select transaction class

Default_TC (Default_SP)

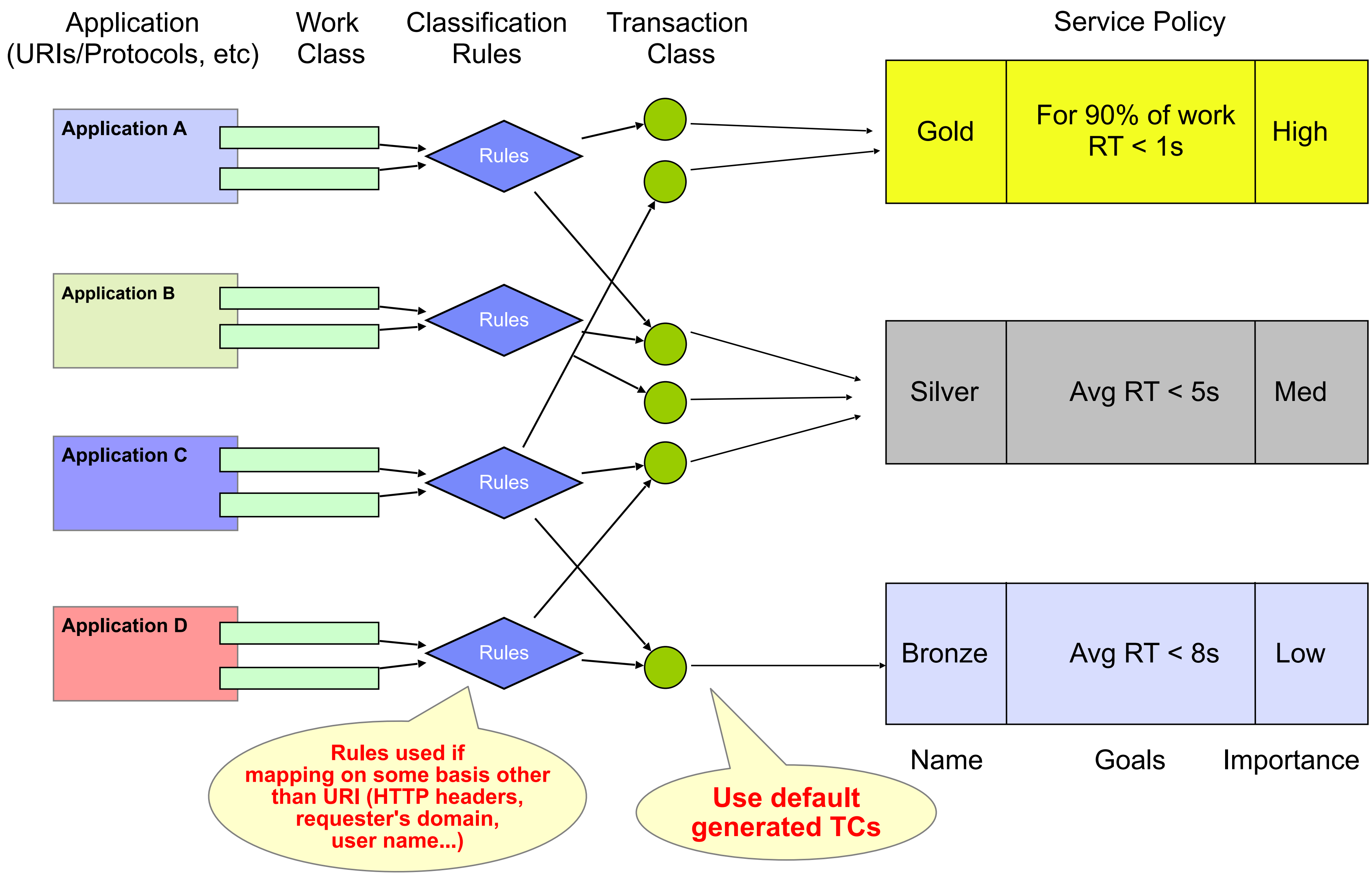
Default_TC (Default_SP)
Default_TC_MyGoldSP (MyGoldSP)

Work classes for SOAP requests

Work classes for IIOP requests

Work classes for JMS requests

Summary of what we just did

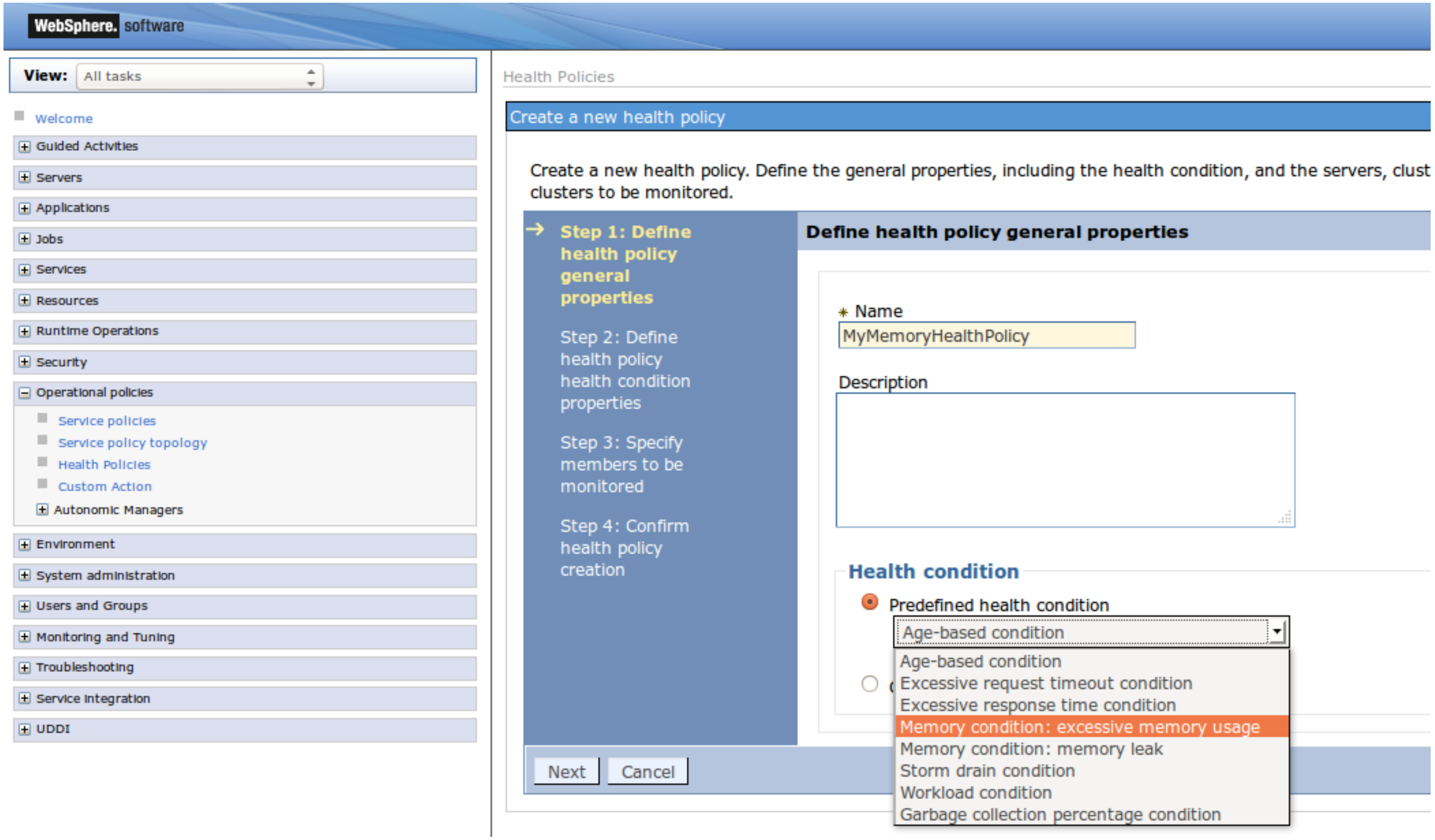


Enabling Health Management for an application

or

How to avoid being woken up in the night!

Creating a health policy



WebSphere. software

View: All tasks

- Welcome
- Guided Activities
- Servers
- Applications
- Jobs
- Services
- Resources
- Runtime Operations
- Security
- Operational policies
  - Service policies
  - Service policy topology
  - Health Policies
  - Custom Action
  - Autonomic Managers
- Environment
- System administration
- Users and Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Health Policies

Create a new health policy

Create a new health policy. Define the general properties, including the health condition, and the servers, clust clusters to be monitored.

→ **Step 1: Define health policy general properties**

Step 2: Define health policy health condition properties

Step 3: Specify members to be monitored

Step 4: Confirm health policy creation

Define health policy general properties

* Name: MyMemoryHealthPolicy

Description:

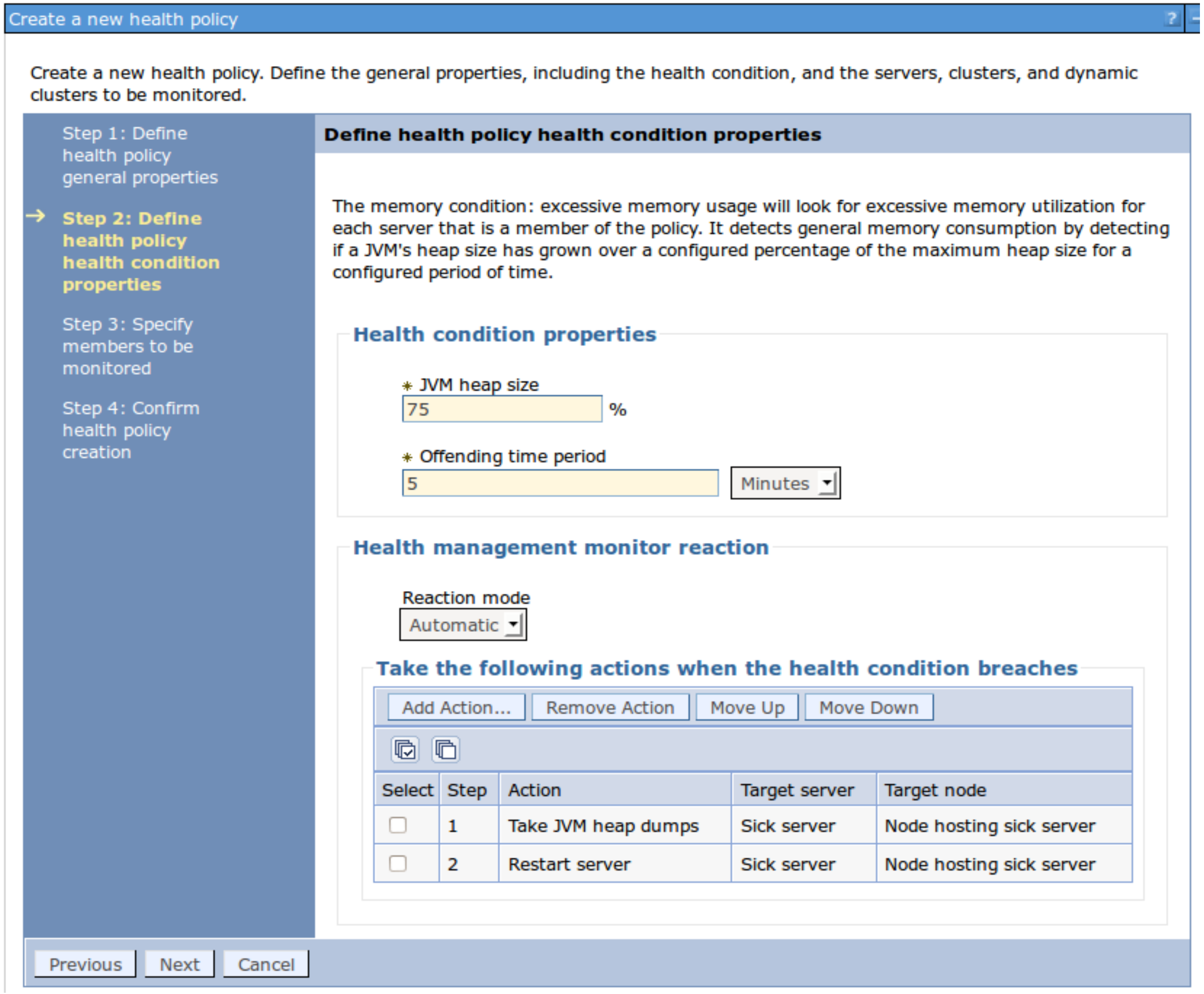
Health condition

● Predefined health condition: Age-based condition

- Age-based condition
- Excessive request timeout condition
- Excessive response time condition
- Memory condition: excessive memory usage
- Memory condition: memory leak
- Storm drain condition
- Workload condition
- Garbage collection percentage condition

Next | Cancel

Excessive memory policy properties



Create a new health policy

Create a new health policy. Define the general properties, including the health condition, and the servers, clusters, and dynamic clusters to be monitored.

Step 1: Define health policy general properties

→ **Step 2: Define health policy health condition properties**

Step 3: Specify members to be monitored

Step 4: Confirm health policy creation

Define health policy health condition properties

The memory condition: excessive memory usage will look for excessive memory utilization for each server that is a member of the policy. It detects general memory consumption by detecting if a JVM's heap size has grown over a configured percentage of the maximum heap size for a configured period of time.

Health condition properties

* JVM heap size: 75 %

* Offending time period: 5 Minutes

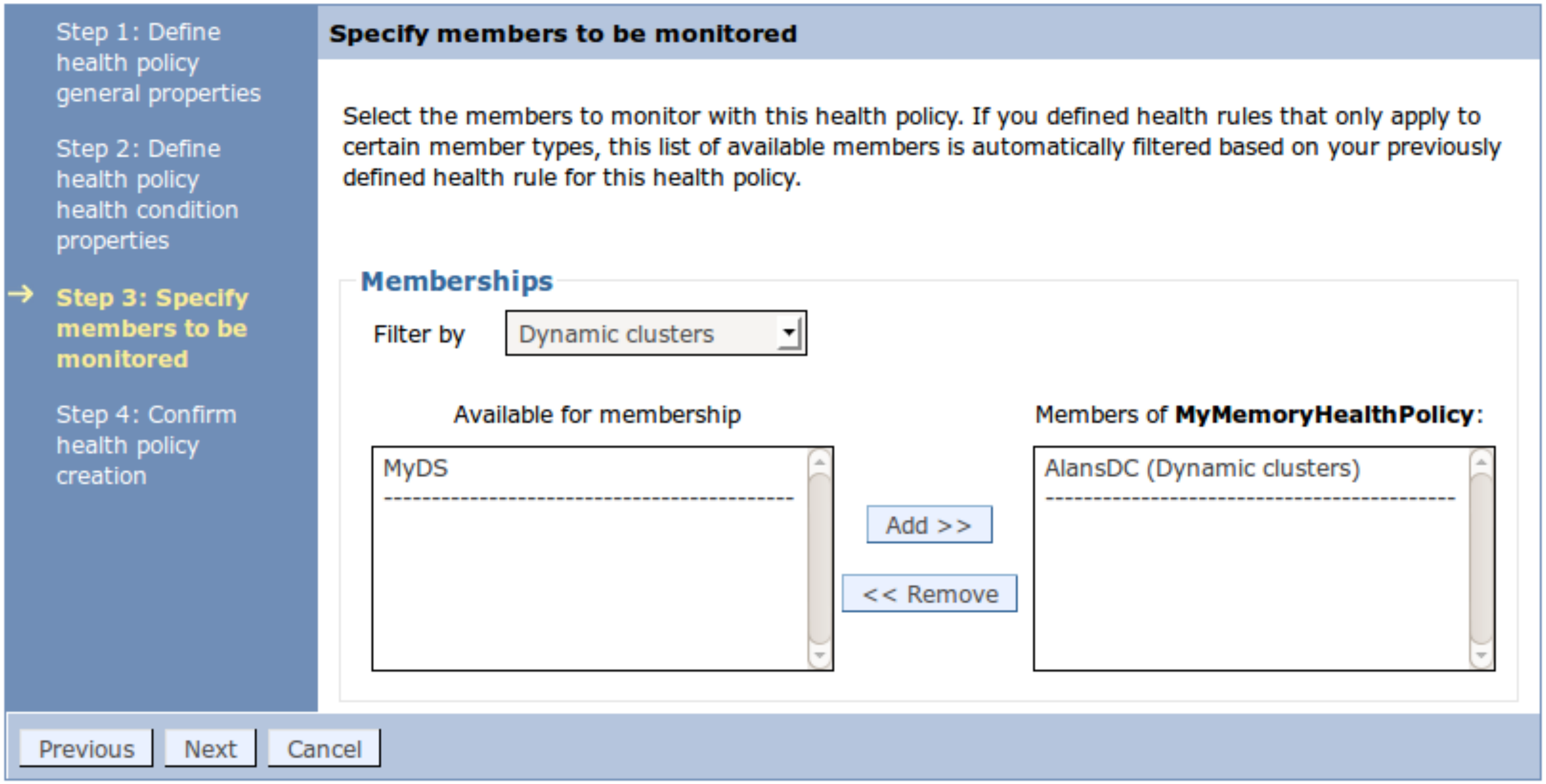
Health management monitor reaction

Reaction mode: Automatic

Take the following actions when the health condition breaches

Select	Step	Action	Target server	Target node
☐	1	Take JVM heap dumps	Sick server	Node hosting sick server
☐	2	Restart server	Sick server	Node hosting sick server

Defining the targets to be monitored



Step 1: Define health policy general properties

Step 2: Define health policy health condition properties

→ **Step 3: Specify members to be monitored**

Step 4: Confirm health policy creation

Specify members to be monitored

Select the members to monitor with this health policy. If you defined health rules that only apply to certain member types, this list of available members is automatically filtered based on your previously defined health rule for this health policy.

Memberships

Filter by: Dynamic clusters

Available for membership: MyDS

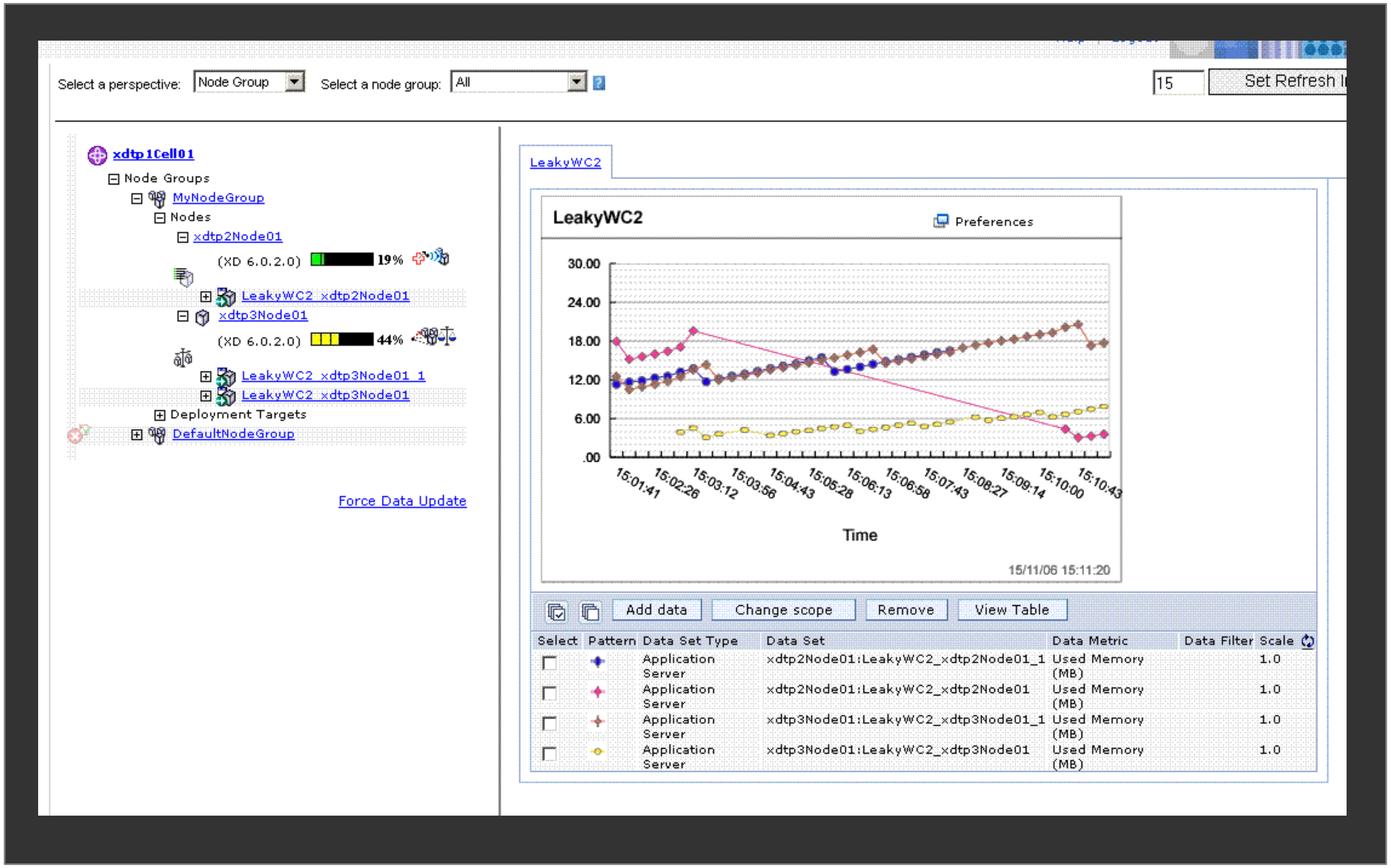
Members of **MyMemoryHealthPolicy**: AlansDC (Dynamic clusters)

The result

- At least one servers in this DC will start automatically
  - as soon as the DC is switched to *Automatic*
- Once it's heap exceeds 75% for 5 mins our health policy is triggered:
  - The heap dump we requested is taken
  - The APC starts (at least) one more member of the DC to handle requests for this application
  - The affected DC member is then restarted
- This carries on forever
  - With no loss of service to users
  - With no need for operator intervention

Don't forget to tell the developers they need to fix the application!

Living with a memory leak



- xdtp1Cell01
  - Node Groups
    - MyNodeGroup
      - Nodes
        - xdtp2Node01
          - (XD 6.0.2.0) 19%
          - LeakyWC2_xdtp2Node01
        - xdtp3Node01
          - (XD 6.0.2.0) 44%
          - LeakyWC2_xdtp3Node01_1
          - LeakyWC2_xdtp3Node01
      - Deployment Targets
    - DefaultNodeGroup

Select	Pattern	Data Set Type	Data Set	Data Metric	Data Filter	Scale
☐	◆	Application Server	xdtp2Node01:LeakyWC2_xdtp2Node01_1	Used Memory (MB)		1.0
☐	◆	Application Server	xdtp2Node01:LeakyWC2_xdtp2Node01	Used Memory (MB)		1.0
☐	◆	Application Server	xdtp3Node01:LeakyWC2_xdtp3Node01_1	Used Memory (MB)		1.0
☐	●	Application Server	xdtp3Node01:LeakyWC2_xdtp3Node01	Used Memory (MB)		1.0

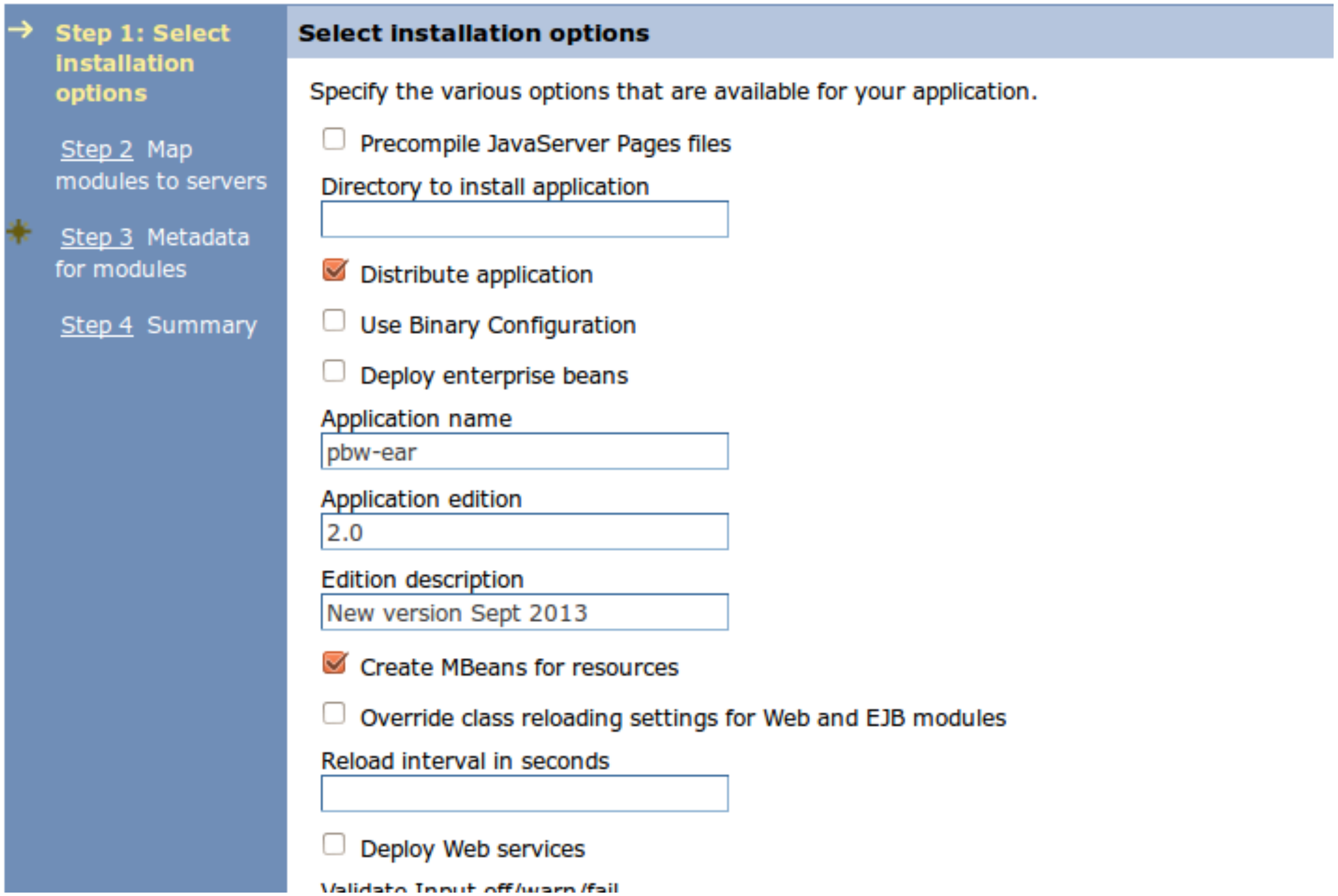
Note: this demonstration was captured using an early version of WebSphere XD which had several cosmetic differences from the current version in WAS 8.5.5

Update your applications seamlessly

or

Using Application Edition Management

Installing a new application edition



→ **Step 1: Select installation options**

Step 2 Map modules to servers

Step 3 Metadata for modules

Step 4 Summary

Select installation options

Specify the various options that are available for your application.

- [ ] Precompile JavaServer Pages files

Directory to install application:

- [✓] Distribute application
- [ ] Use Binary Configuration
- [ ] Deploy enterprise beans

Application name: pbw-ear

Application edition: 2.0

Edition description: New version Sept 2013

- [✓] Create MBeans for resources
- [ ] Override class reloading settings for Web and EJB modules

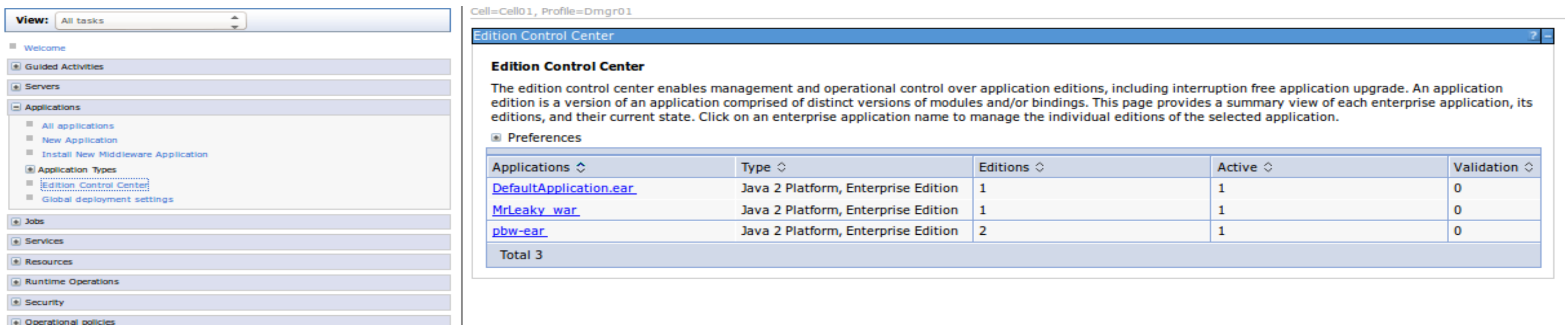
Reload interval in seconds:

- [ ] Deploy Web services

Validate Input off/warn/fail

- Install the application in the usual way
  - EAR file name and everything else are the same
  - Only difference is specifying an edition as shown

Application Edition Manager



Cell=Cell01, Profile=Dmgr01

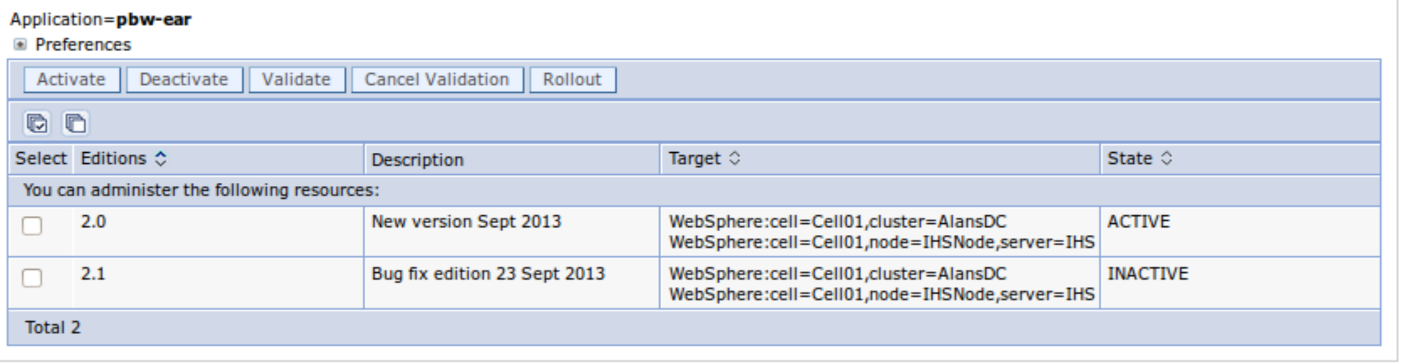
Edition Control Center

Edition Control Center

The edition control center enables management and operational control over application editions, including interruption free application upgrade. An application edition is a version of an application comprised of distinct versions of modules and/or bindings. This page provides a summary view of each enterprise application, its editions, and their current state. Click on an enterprise application name to manage the individual editions of the selected application.

Preferences

Applications	Type	Editions	Active	Validation
DefaultApplication.ear	Java 2 Platform, Enterprise Edition	1	1	0
MrLeaky_war	Java 2 Platform, Enterprise Edition	1	1	0
pbw-ear	Java 2 Platform, Enterprise Edition	2	1	0
Total 3				

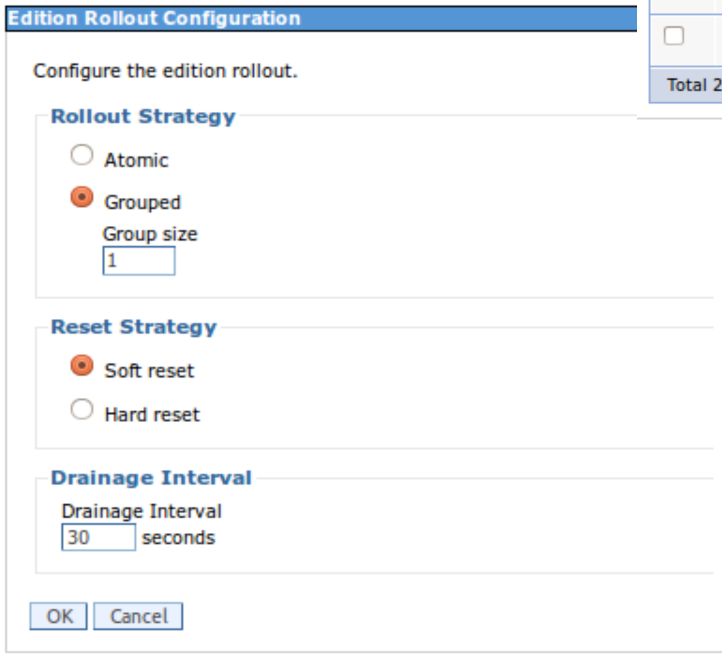


Application=**pbw-ear**

Preferences

You can administer the following resources:

Select	Editions	Description	Target	State
☐	2.0	New version Sept 2013	WebSphere:cell=Cell01,cluster=AlansDC WebSphere:cell=Cell01,node=IHSNode,server=IHS	ACTIVE
☐	2.1	Bug fix edition 23 Sept 2013	WebSphere:cell=Cell01,cluster=AlansDC WebSphere:cell=Cell01,node=IHSNode,server=IHS	INACTIVE
Total 2				



Edition Rollout Configuration

Configure the edition rollout.

Rollout Strategy

Atomic [✓ Grouped]

Group size: 1

Reset Strategy

[✓ Soft reset] Hard reset

Drainage Interval

Drainage Interval: 30 seconds

- Click the application to see list and status of editions
- Select an inactive edition and click “Rollout” to make it active
  - *Grouped: least disruptive to service*
  - *Atomic: ensures no overlap between old and new editions*

In summary...

WAS 8.5.5 Intelligent Management gives you:

- Performance Management
  - Save money by running servers at higher utilisation
  - Make adding and removing capacity easier and less finger-in-the-air
- Health Management
  - Provide more resilient service to users
  - Reduce support/operations costs
- Application Edition Management
  - More productive application validation and upgrades

The savings and benefits are large for no cost and minimal change to your configuration

It's much too valuable to ignore!