WebSphere Portal Advanced Administration and High Availability Infrastructure Design

Johnny Teoh
jteoh@au1.ibm.com

Certified Senior Client Technical Professional
IBM SWG Lotus, Asia Pacific

Agenda

- Portal Operational Architecture
- Recap on changes between Portal 6.0 and 6.1
- Portal Command Line Interface
- Other Portal Deployment Concerns
- Overview of Portal Cluster Setup
- Troubleshoot of Portal Cluster
- Portal Security Overview
  - Portal Authentication
  - Portal Authorisation

Portal Operational Architecture

Factors that can Affect the Operational Architecture

- How many applications to expect?
- And there multiple geographical locations?
- What is the expected size of the site at saturation point?
- Does the site need to be available 24 x 7?
- How many authors will concurrently access the system?
- How often will the content be updated?
- Can the site be personalized or targeted to groups of users?
- Usage distribution between the backend systems

What is Clustering?

- Portal clusters are based on WebSphere Application Server clustering.
- Clusters are a set of application servers (WebSphere_Portal) with the same applications installed and grouped logically for:
  - Ease of administration
  - Workload management
- Applications installed to the cluster are automatically distributed to the cluster members.
- There are several cluster configurations
  - Vertical scaling
  - Horizontal scaling
  - Combination of vertical and horizontal scaling

Software Stack Options in a Typical Portal Setup

1. Tier

- HTTP Server
- Edge Components
- Caching Proxy
- Security Proxy

2. Tier

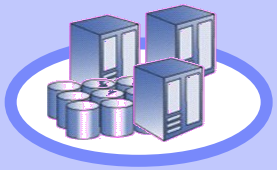
- WebSphere Portal
- Content Apps
- Search Server

3. Tier

- Database Server

Backend

- Domino
- Lotus IM and Web Conferencing
- Connections
- Quickr
- WebServices
- WSRP Services

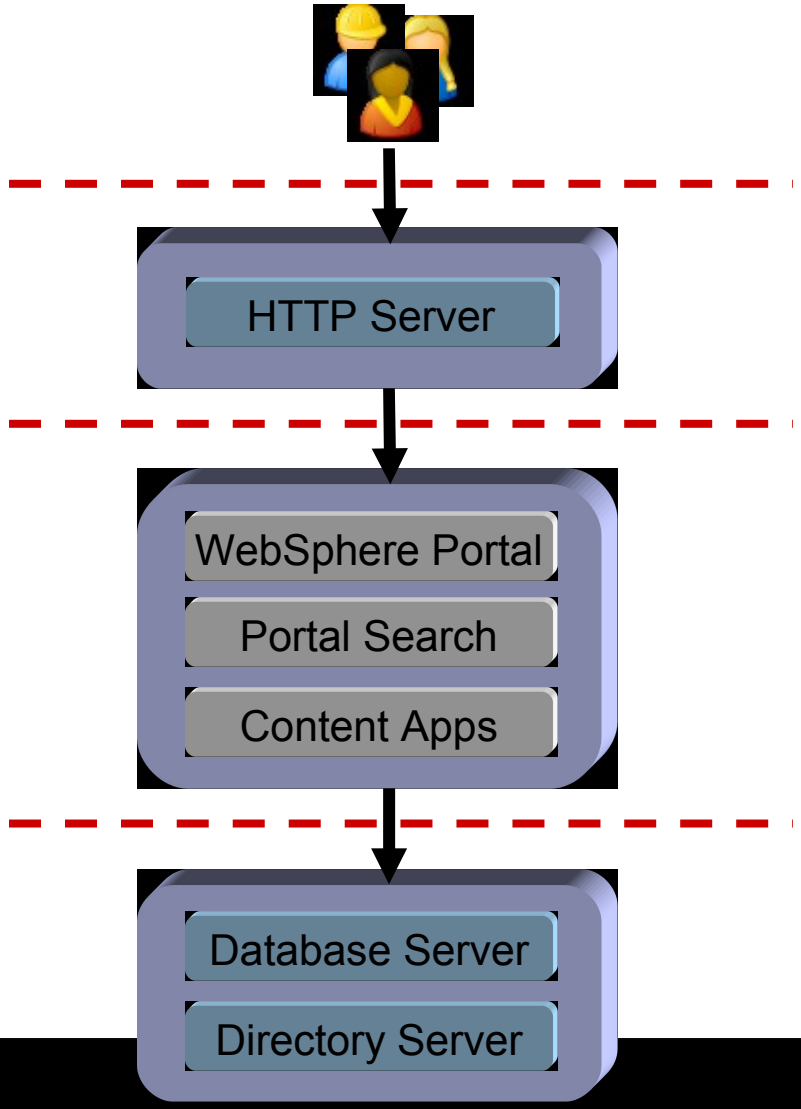


Management

- Directory Server
- Deployment Mgr
- Security Server
- Intelligent Orchestrator

The Pilot Portal

This is the standard for WebSphere Portal running without a cluster.

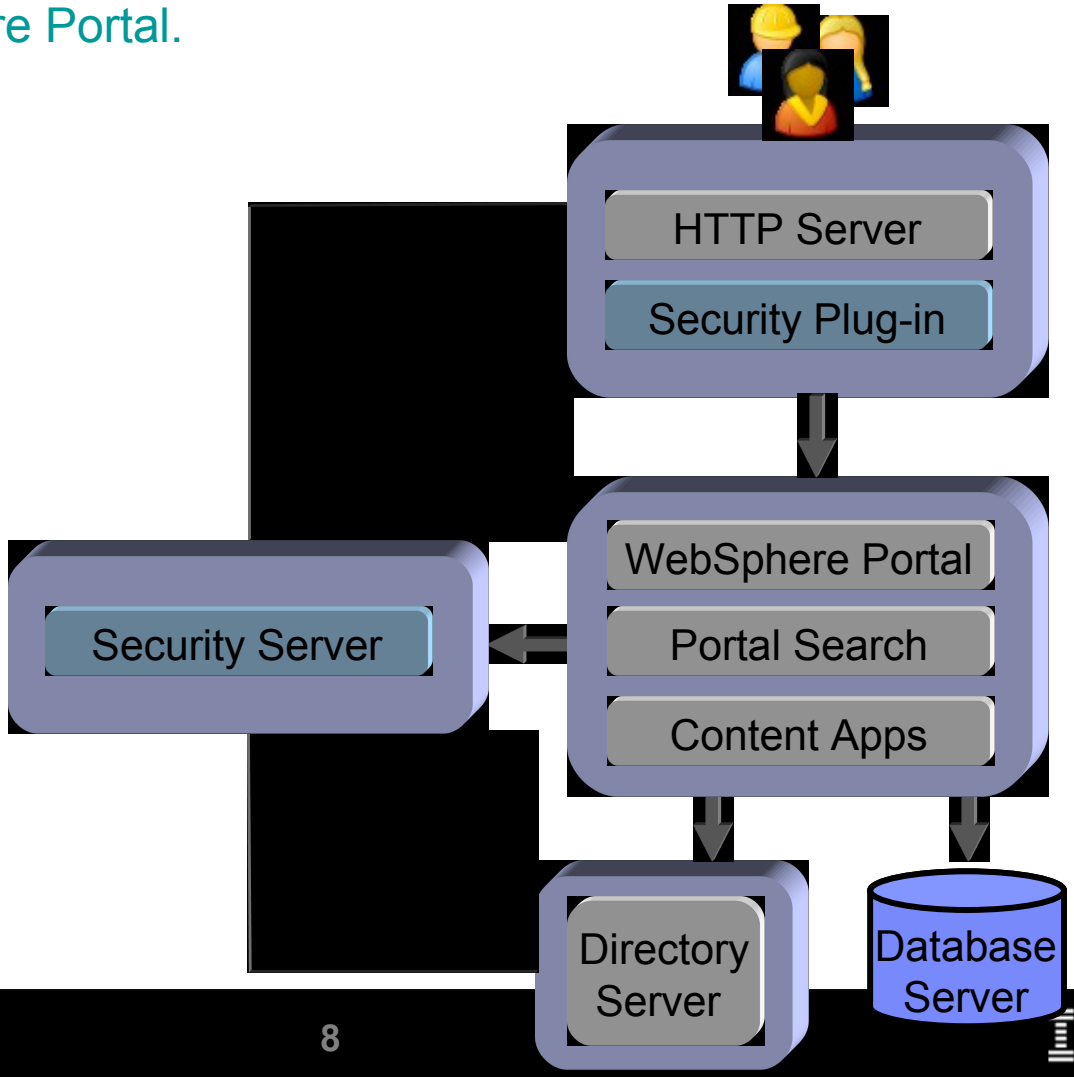


Still simple, but allows flexible placement on nodes to exploit firewall security.

Enhanced Security Portal – Access Manager

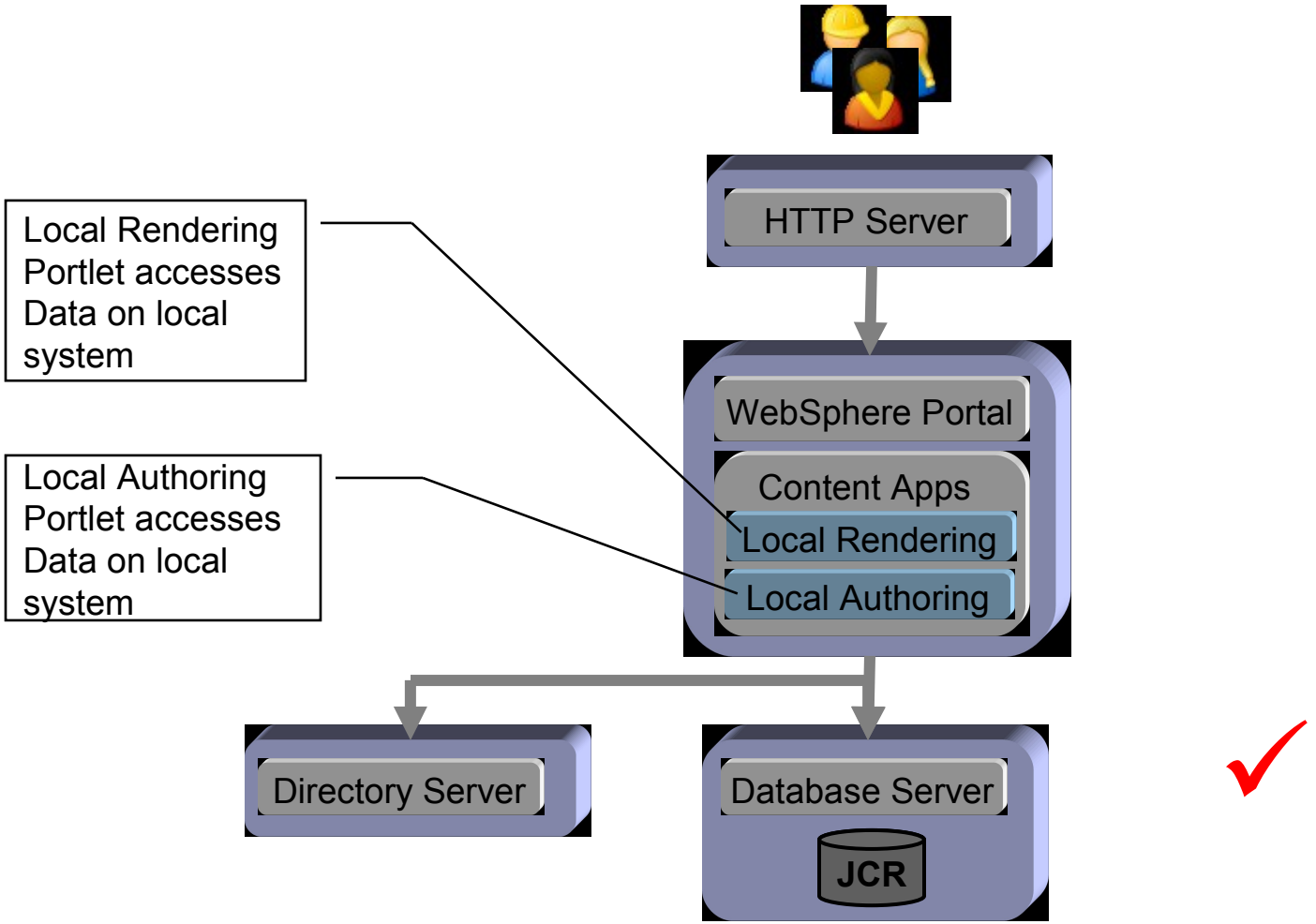
WebSphere Portal supports IBM Tivoli Access Manager for e-business and Computer Associates eTrust® SiteMinder®. There are more products which Support WebSphere Portal.

Split of responsibilities by externalized security management.
- Authentication
- Authorisation
- Credential Store

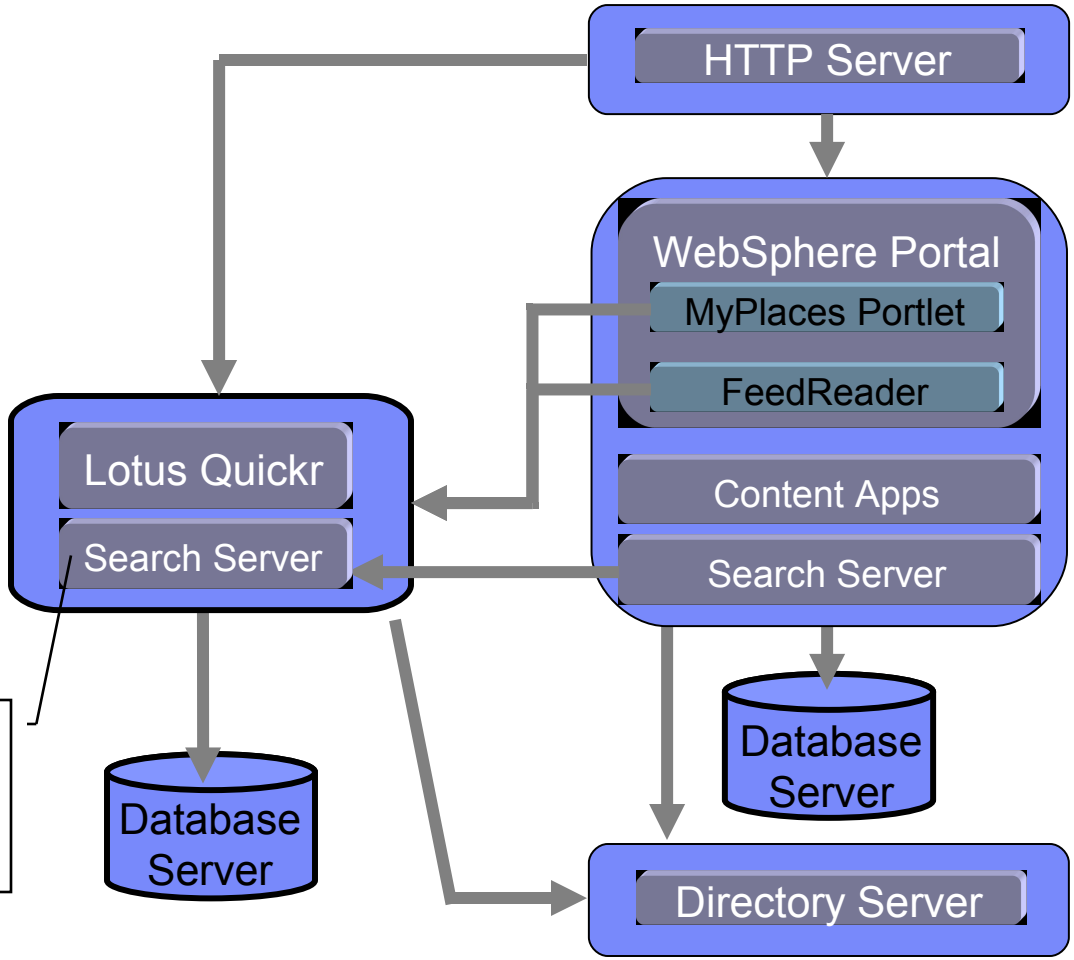


Adding Web Content Management

WCM on a single server



Inclusion of Lotus Quickr

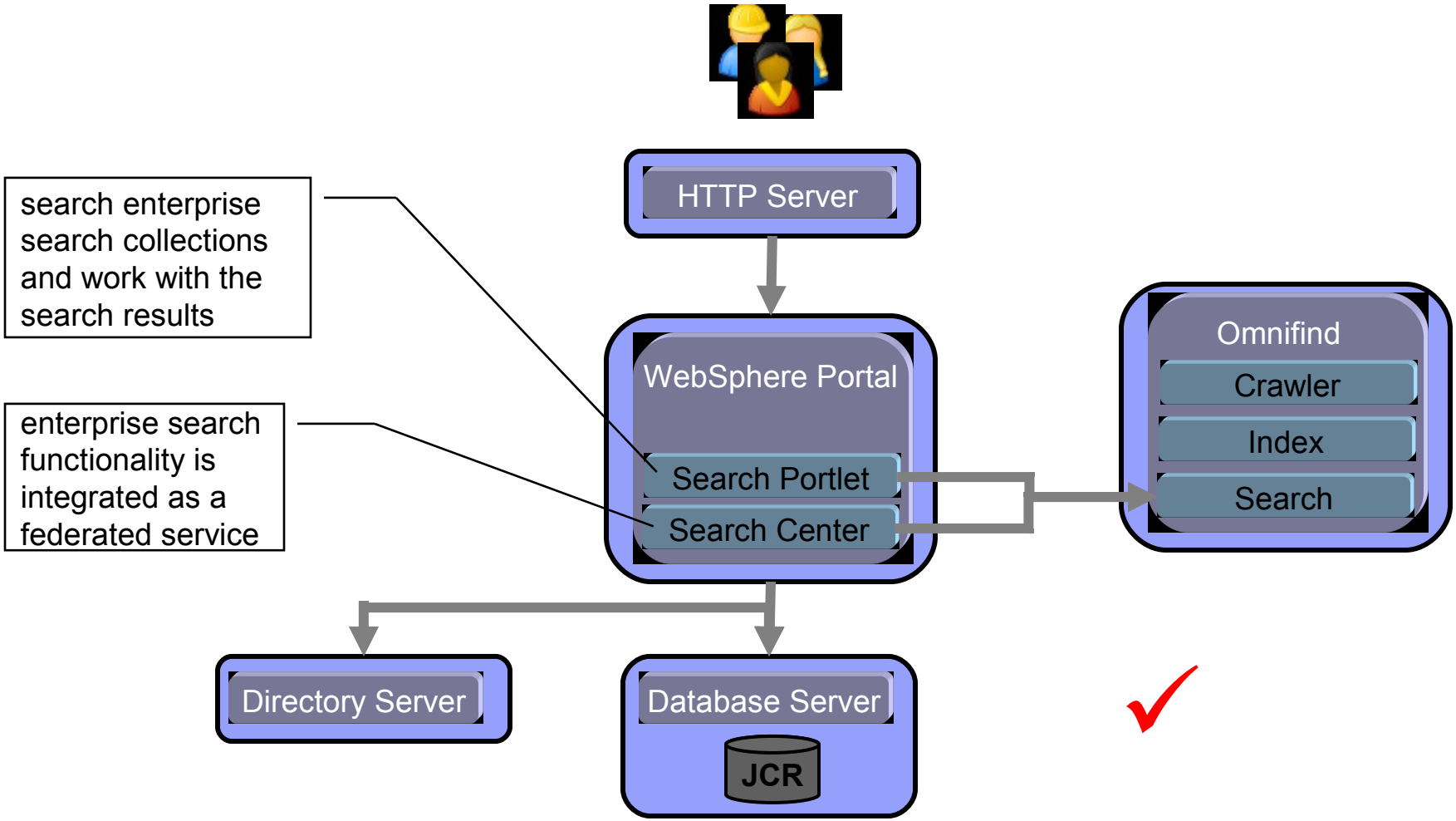


Single-Signon (SSO) is required to search across Quickr information

To configure the search see :
http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1m0/index.jsp?topic=/com.ibm.wp.ent.doc/admin/srtprprscrrmtsrchsrv.html

Adding Omnifind single Server

Portal and Omnifind single server deployments



Content Accelerator

PUT INFORMATION TO WORK WITH WEB CONTENT MANAGEMENT AND SEARCH

IBM Lotus Web Content Management software enables companies to rapidly build next generation Web sites with tools that put the power of content creation into the hands of line of business, subject matter experts and enhancements that can be used throughout the entire content lifecycle.

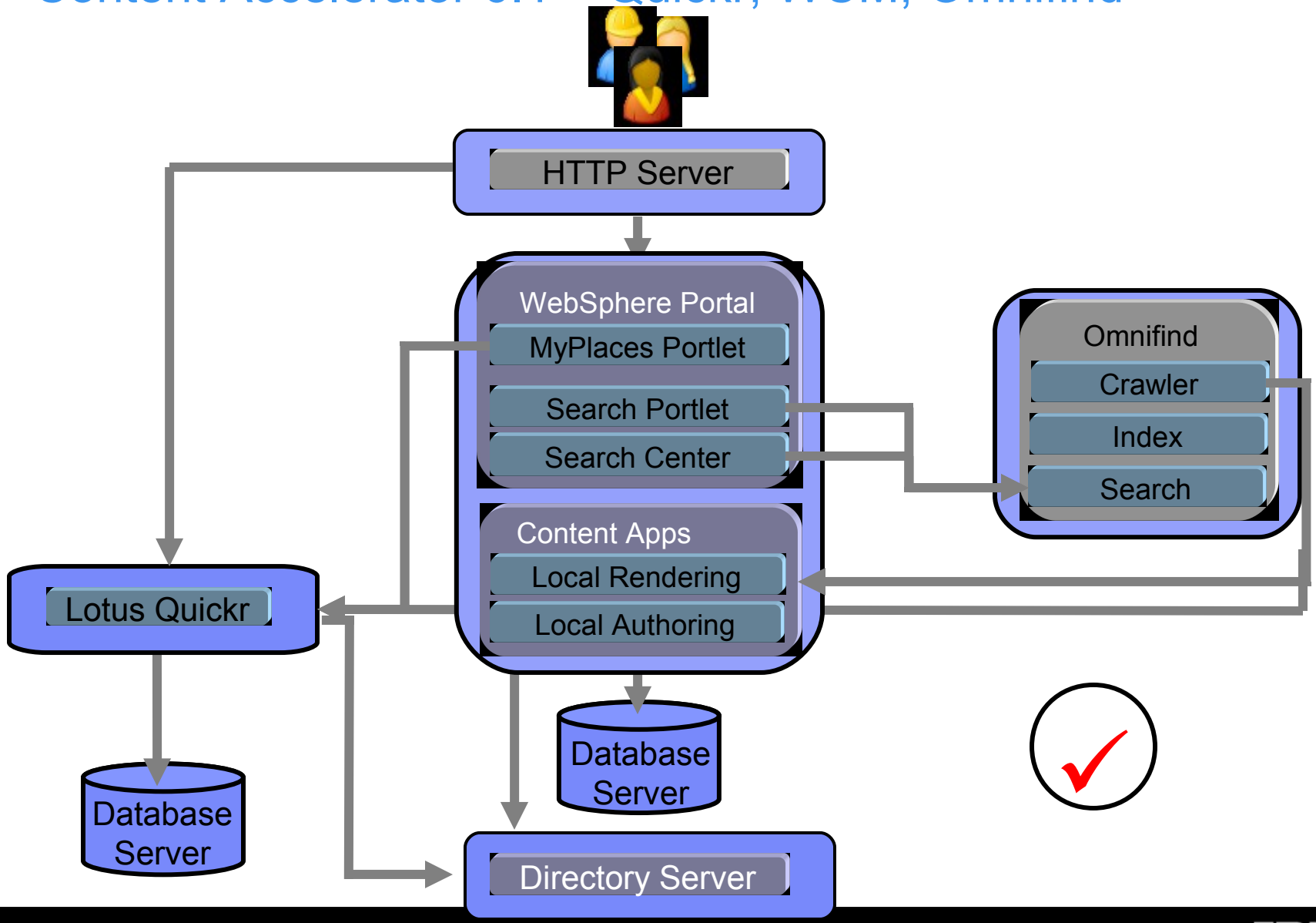
IBM Lotus® Quickr™ is a Web 2.0-based offering that helps teams be more effective and deliver more results. People can share information, documents, and even rich media — with colleagues, customers, and business partners — whether they are inside or outside the firewall.*

IBM OmniFind™ Enterprise Edition software delivers high-quality, scalable and security-rich enterprise search to maximize the value of corporate information**

** Licensed to use Doucment Library Places (J2EE), no connectors*
*** 1 Test and 1 Production Processor Core*

Content Accelerator Wiki:
http://www-10.lotus.com/ldd/portalwiki.nsf/archive?openview&title=IBM%20Content%20Accelerator&type=cat&cat=IBM%20Content%20Accelerator&sort=l

Content Accelerator 6.1 – Quickr, WCM, Omnifind



Collaboration Accelerator 6.1

IBM Lotus® Sametime® software provides instant, anytime access to people and information through three on demand concepts: presence awareness, business instant messaging and Web conferencing.*

IBM Lotus® Quickr™ is a Web 2.0-based offering that helps teams be more effective and deliver more results. People can share information, documents, and even rich media — with colleagues, customers, and business partners — whether they are inside or outside the firewall.*

IBM Lotus Connections software is a Web 2.0-based platform for business-grade social computing. It features Web 2.0-based components — Activities, Communities, Dogear (social tagging), Profiles and Blogs — that help people find expertise, share insights, and build new relationships based on their business needs.*

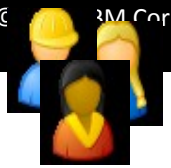
Collaboration Accelerator Wiki:
http://www-10.lotus.com/ldd/portalwiki.nsf/archive?openview&title=IBM%20Collaboration%20Accelerator&type=cat&cat=IBM%20Collaboration%20Accelerator&sort=l

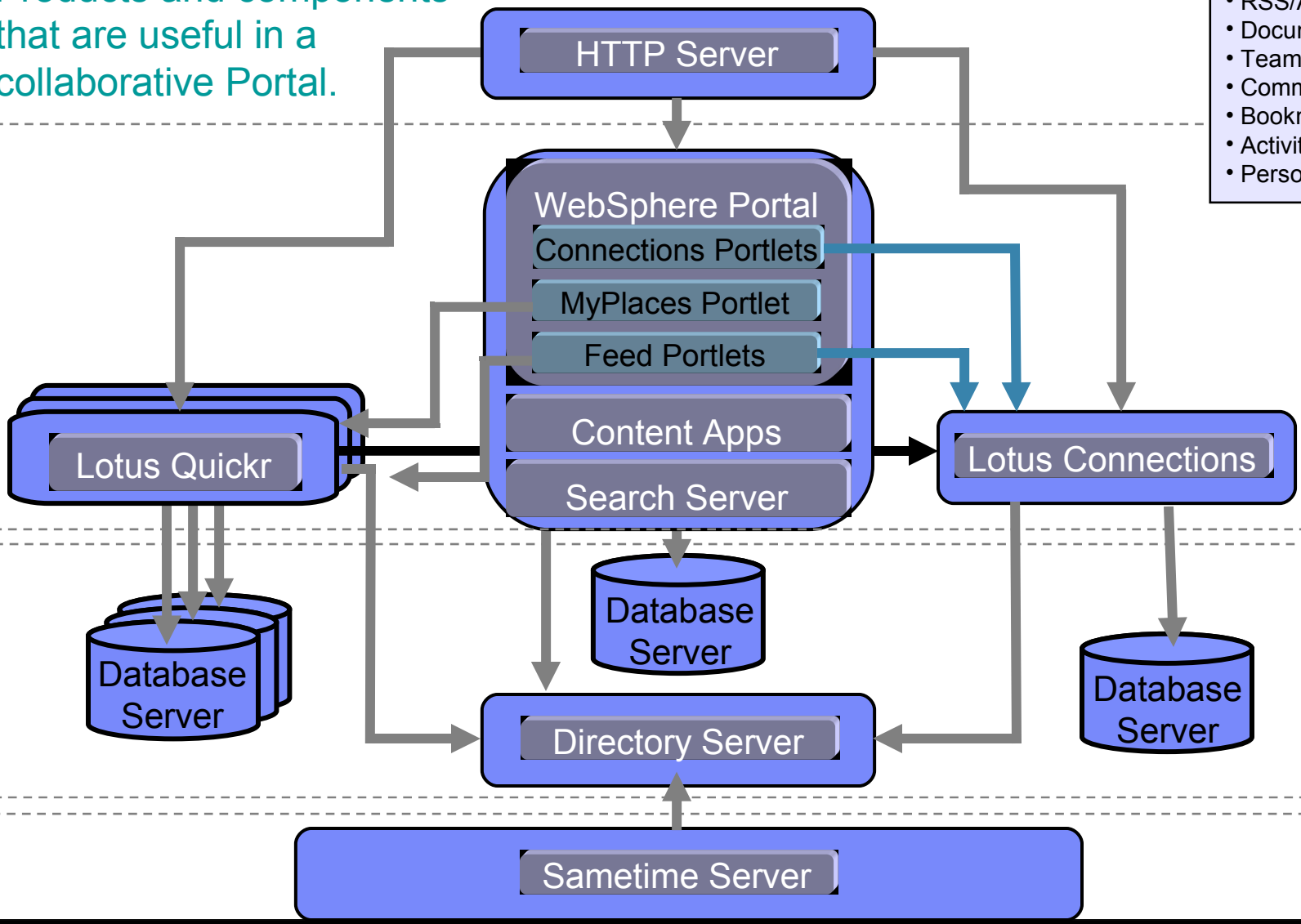
** Browser based access (J2EE services).*

Collaboration Portal

Products and components that are useful in a collaborative Portal.

Collaboration through:
- Blogs, WIKIs
- RSS/ATOM feeds
- Document Libraries
- Team Spaces
- Communities
- Bookmarks
- Activities
- Personal Profiles

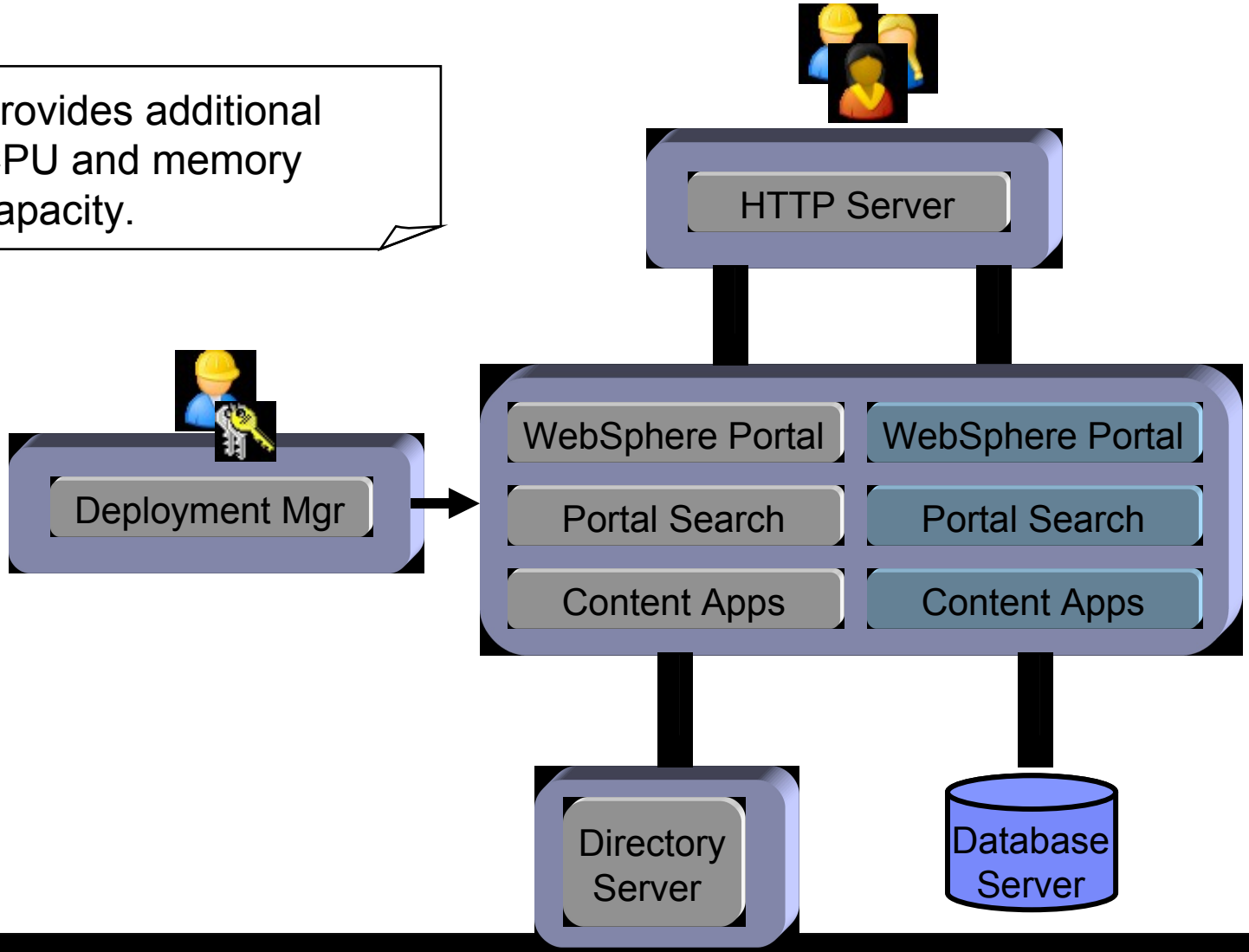




The Basic Vertical Portal Cluster

Using WebSphere Application Server Clustering capabilities.

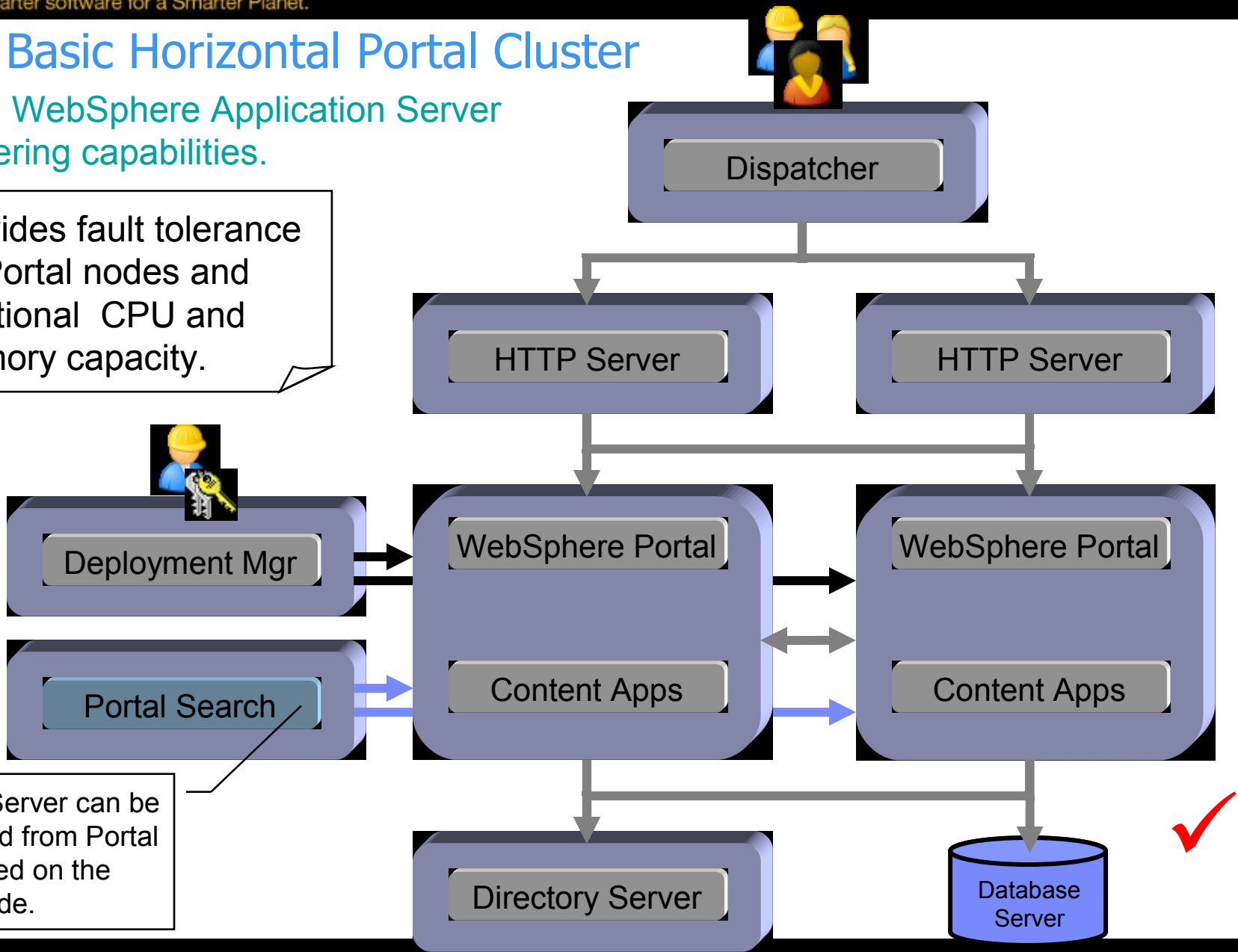
Provides additional CPU and memory capacity.



The Basic Horizontal Portal Cluster

Using WebSphere Application Server Clustering capabilities.

Provides fault tolerance for Portal nodes and additional CPU and memory capacity.



Search Server can be separated from Portal or installed on the same node.

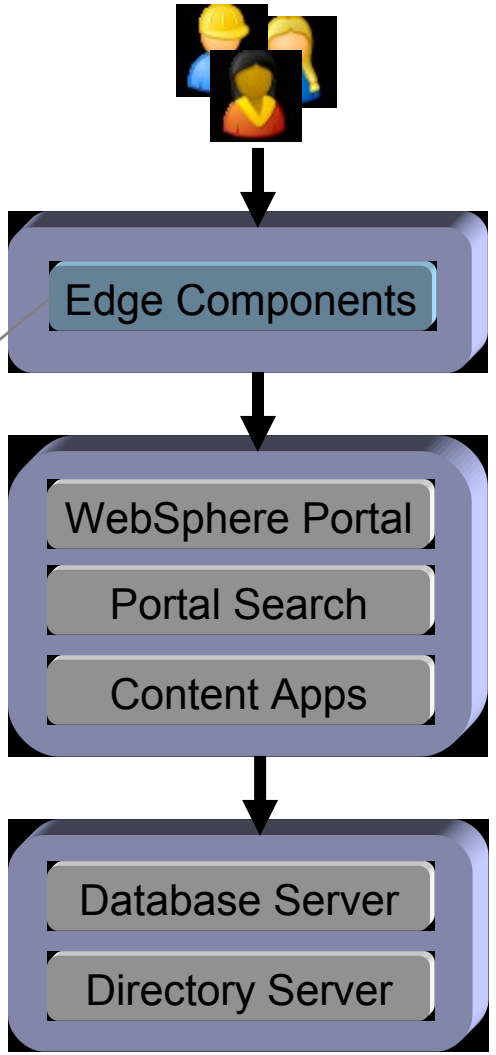
Portal at the Edge

Using caching capabilities for your WebSphere Portal. (e.g. reverse caching proxies)

The cache allows you to serve the following content from cache instead of Portal Server:

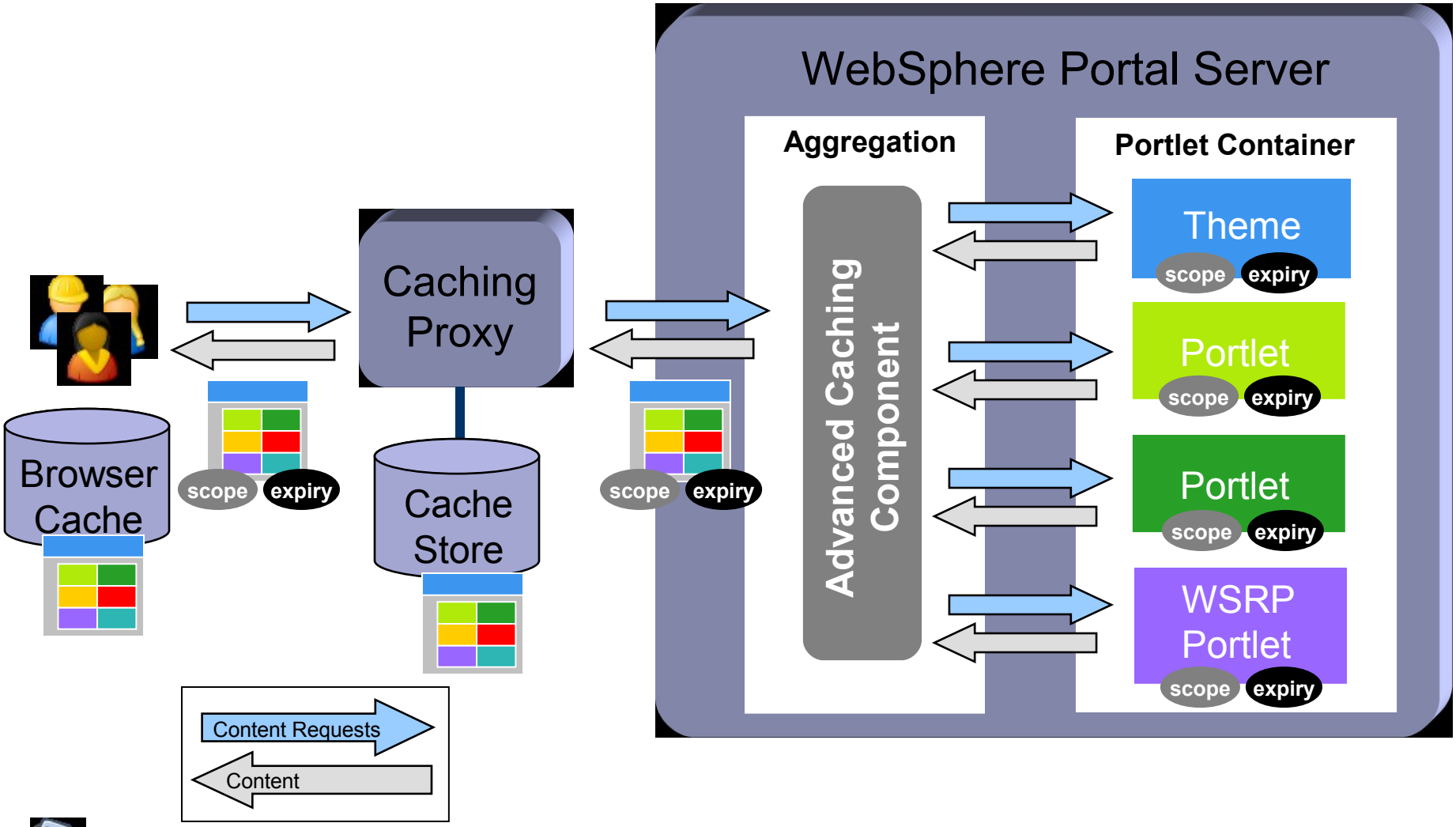
- All kinds static content (HTML Pages, Images, Stylesheets, Java Script libraries,….)
- Information centric Portal Pages

- Caching Proxy
- Content Based Routing (CBR)
- Security Plugin



WhitePaper: www.ibm.com/developerworks/websphere/techjournal/0506_liesche/0506_liesche.html
Edge Components (CBR): www-306.ibm.com/software/webservers/appserv/doc/v51/ec/infocenter/edge/LBguide.htm#HDRCBRWAS
Configuraztion Hints: http://www-1.ibm.com/support/docview.wss?rs=688&uid=swg21266743

WebSphere Portal - Page Caching Overview



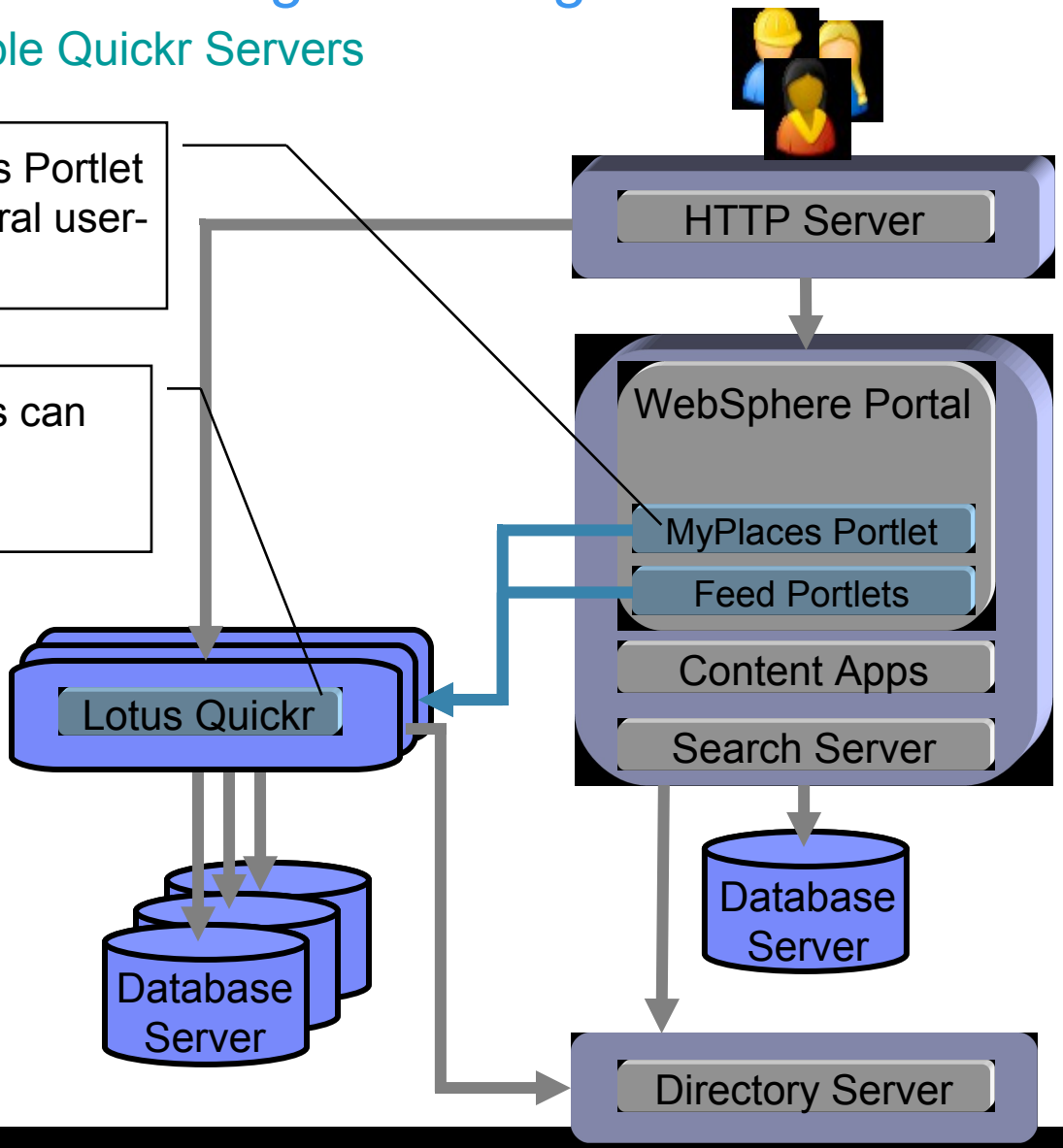
http://www.ibm.com/developerworks/websphere/techjournal/0506_liesche/0506_liesche.html

Analysis for Scaling and Usage

Allow multiple Quickr Servers

The MyPlaces Portlet provides central user-access.

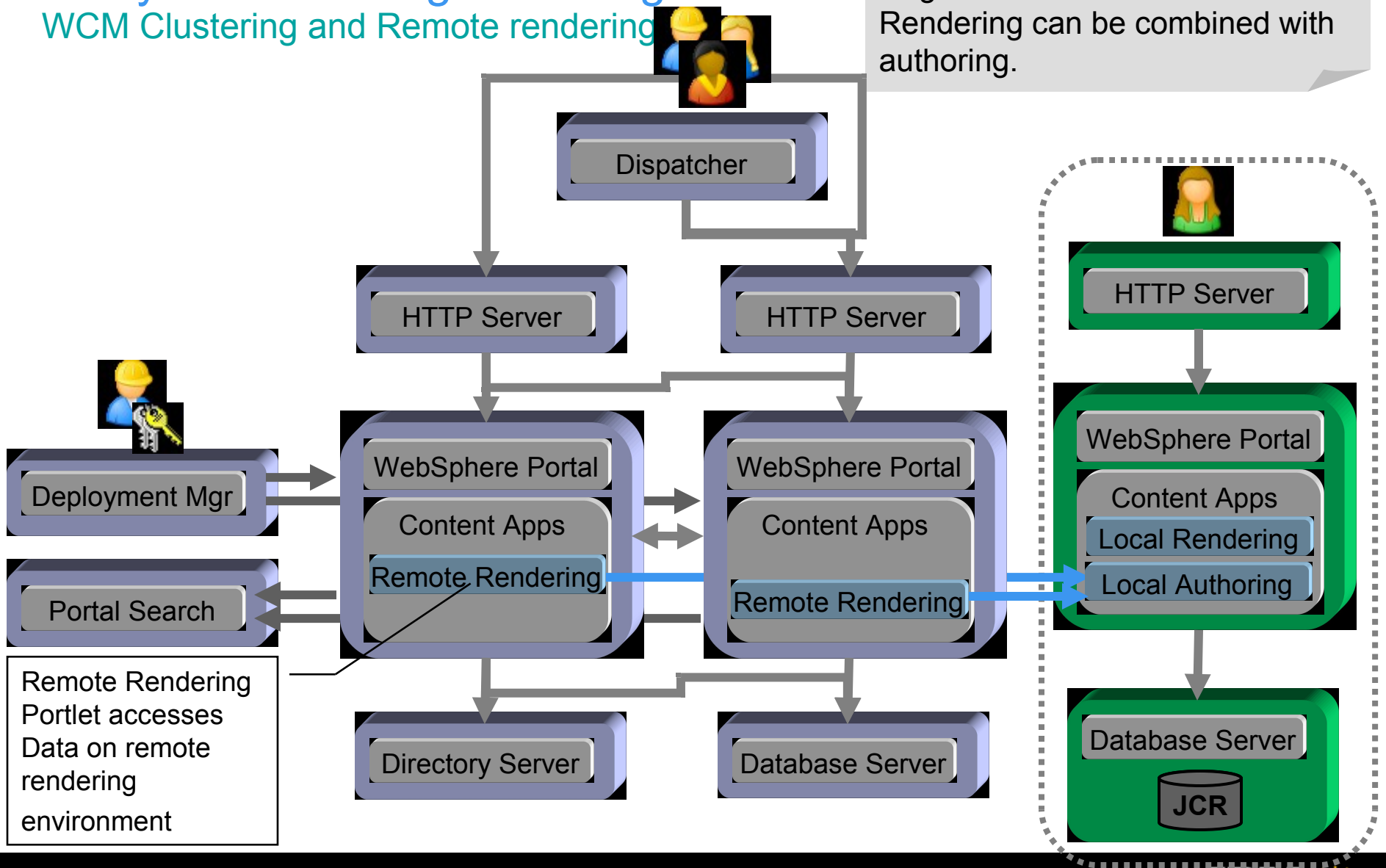
Quickr Servers can be scaled independently.



Analysis for Scaling and Usage

WCM Clustering and Remote rendering

Remote rendering servers can be single nodes or clusters. Rendering can be combined with authoring.

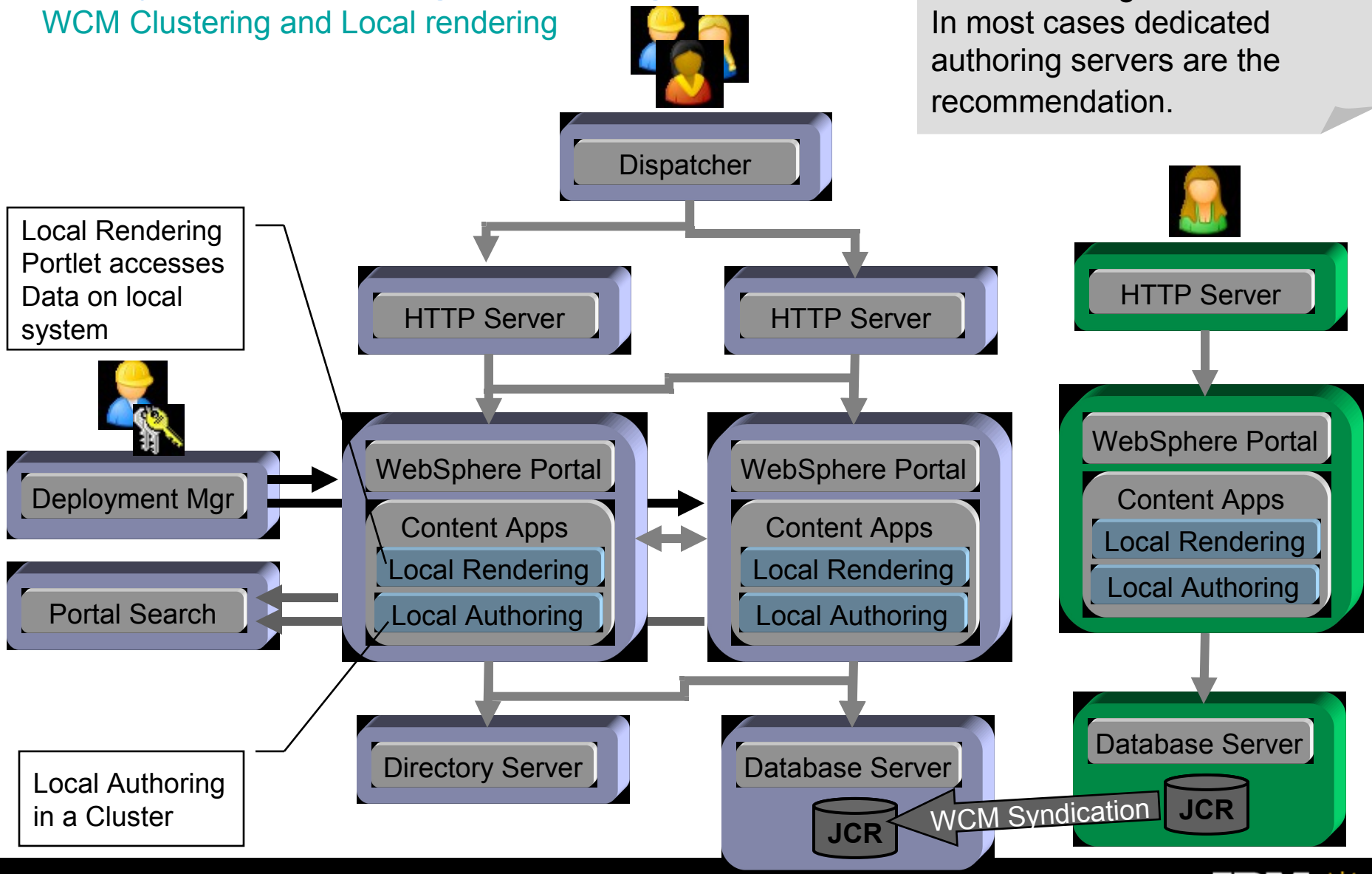


Remote Rendering Portlet accesses Data on remote rendering environment

Analysis for Scaling and Usage

WCM Clustering and Local rendering

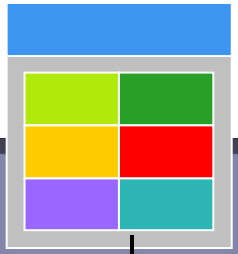
Dedicated authoring servers or local authoring alternatives. In most cases dedicated authoring servers are the recommendation.



Analysis for Workload Distribution/Isolation

Remote Portlets (WSRP)

Portal Users



WebServices for Remote Portlets
(WSRP 2.0)

WebSphere Portal

Portal Aggregation

Portlet Container
Classic / JSR 286

Portal Configuration

WS Stack

Portlet Server
JVM 1

WebSphecre
Application Server
(WAS) 6.1

Portlet Container
JSR 168

Portlet Server
JVM 2

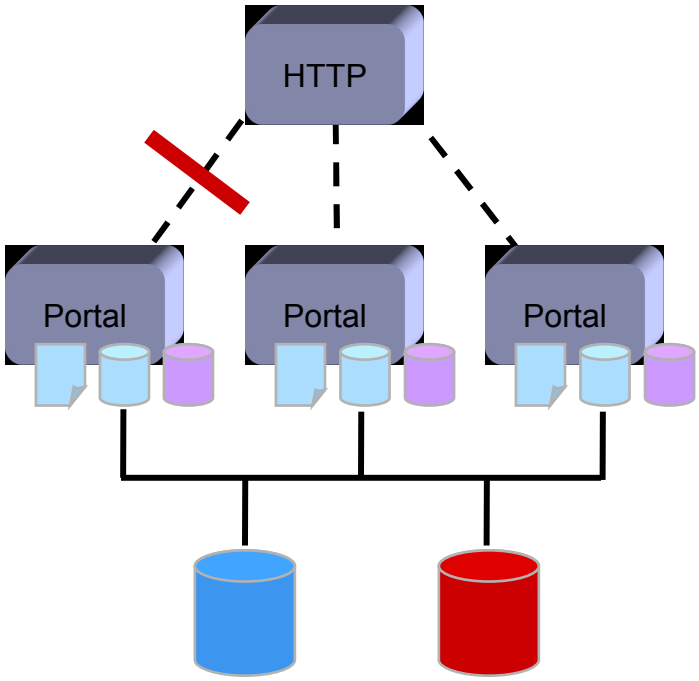
WebSphere Portal
or WAS 7

Portlet Container
Classic / JSR 286

Analysis for Availability and Maintenance

with Data Separation .

Data Separation



1. Take offline
2. Update
3. Test via port xxx
4. Put back on line
5. Repeat…

Partition Portal data into logical domains

- **Config data:** things like class files and property files. Things that are rightfully managed by WAS ND
- **Release data:** portal pages and the portlet instances on those pages, including edit default mode data
- **JCR Data:** WCM content, personalization rules
- **Customization data:** portlet data (user data), page customizations, nothing that is shared
- **Community data:** application instances, etc. Shared data that isn't part of the release data.

with user customization and shared content

Exploring the Portal Configuration Database

- Release database
  - Portal content definitions, rules, rights
- Customisation database
  - Data associated with a particular user only, eg: Portlet preferences
- Community database
  - Resources modified during production, eg: application resources
- JCR database
  - Personalisation data
- Personalisation feedback database
  - Logging of site usage
- LikeMinds database
  - Transactional and ratings data for personalisation

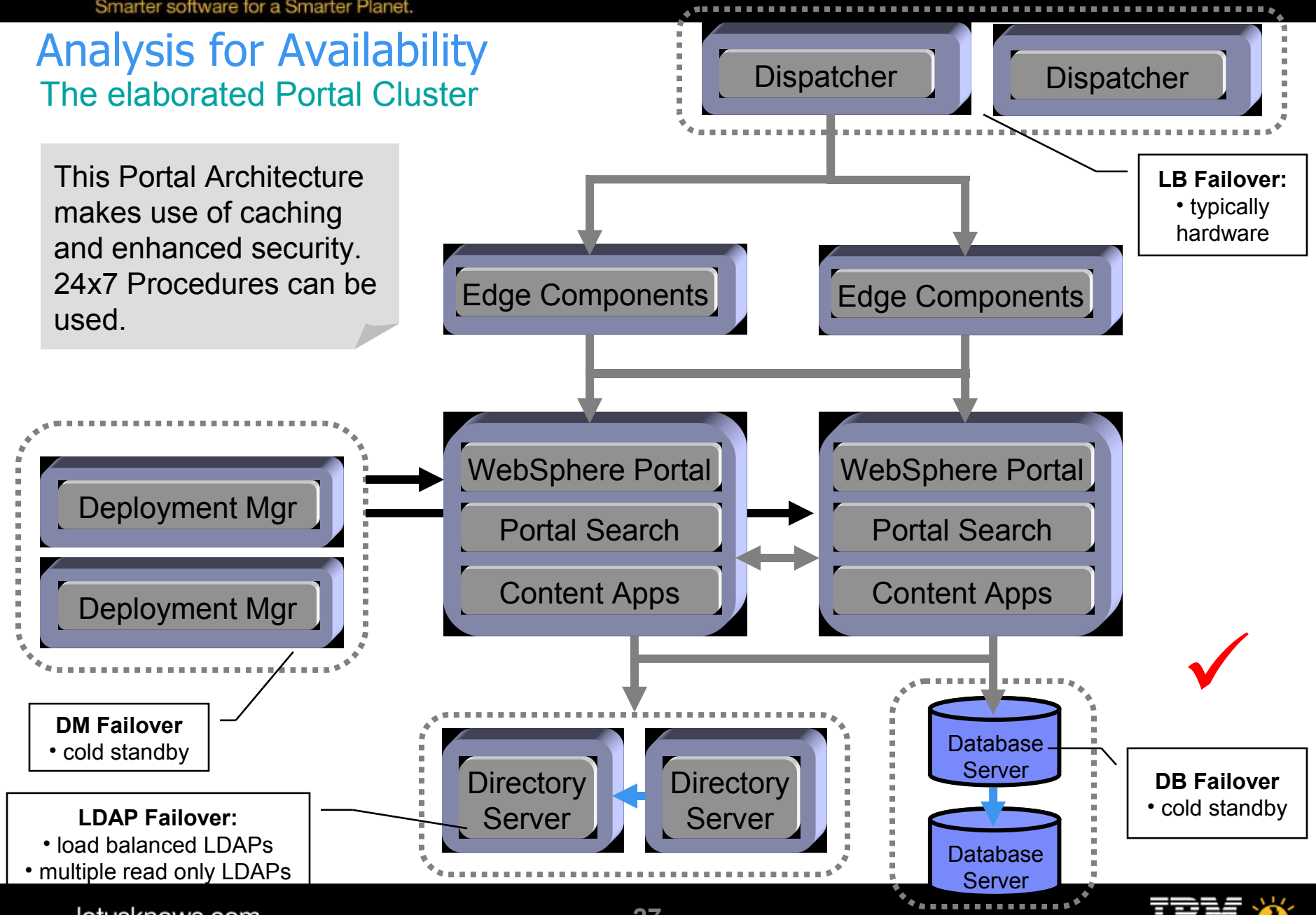
Splitting the Portal Configuration Database

- Multiple Schemas used in each instance of WebSphere Portal
- Options:
  - Keep as one database
  - Split certain schemas into their own databases
  - Split all schemas into separate databases
  - Split databases across multiple server
- Deciding to split:
  - Performance: splitting into multiple databases may improve performance
  - Split to allow replication and redundancy of databases
  - Run the portal without the customisation database (eg: during disaster)

Analysis for Availability

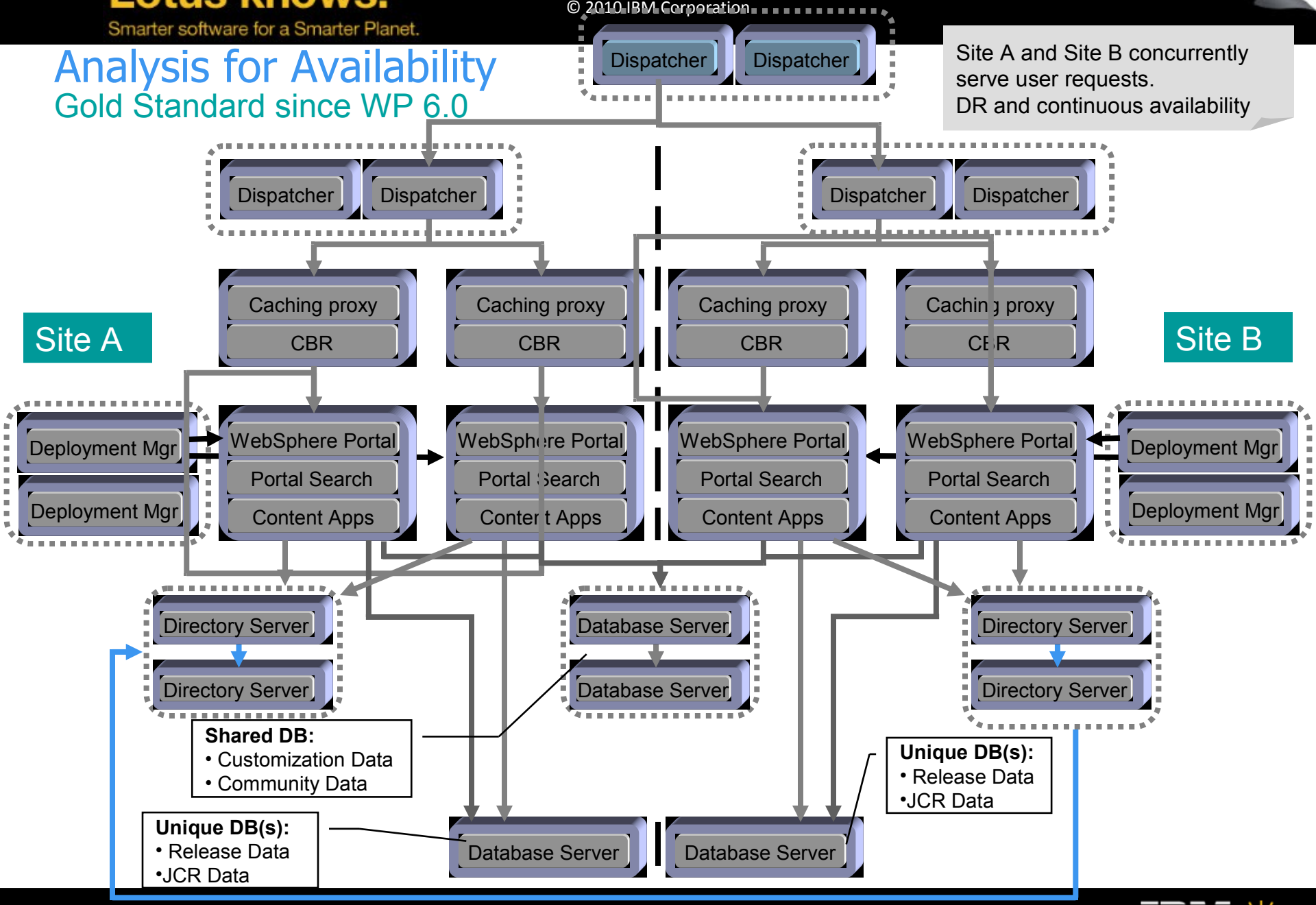
The elaborated Portal Cluster

This Portal Architecture makes use of caching and enhanced security. 24x7 Procedures can be used.



Analysis for Availability
Gold Standard since WP 6.0

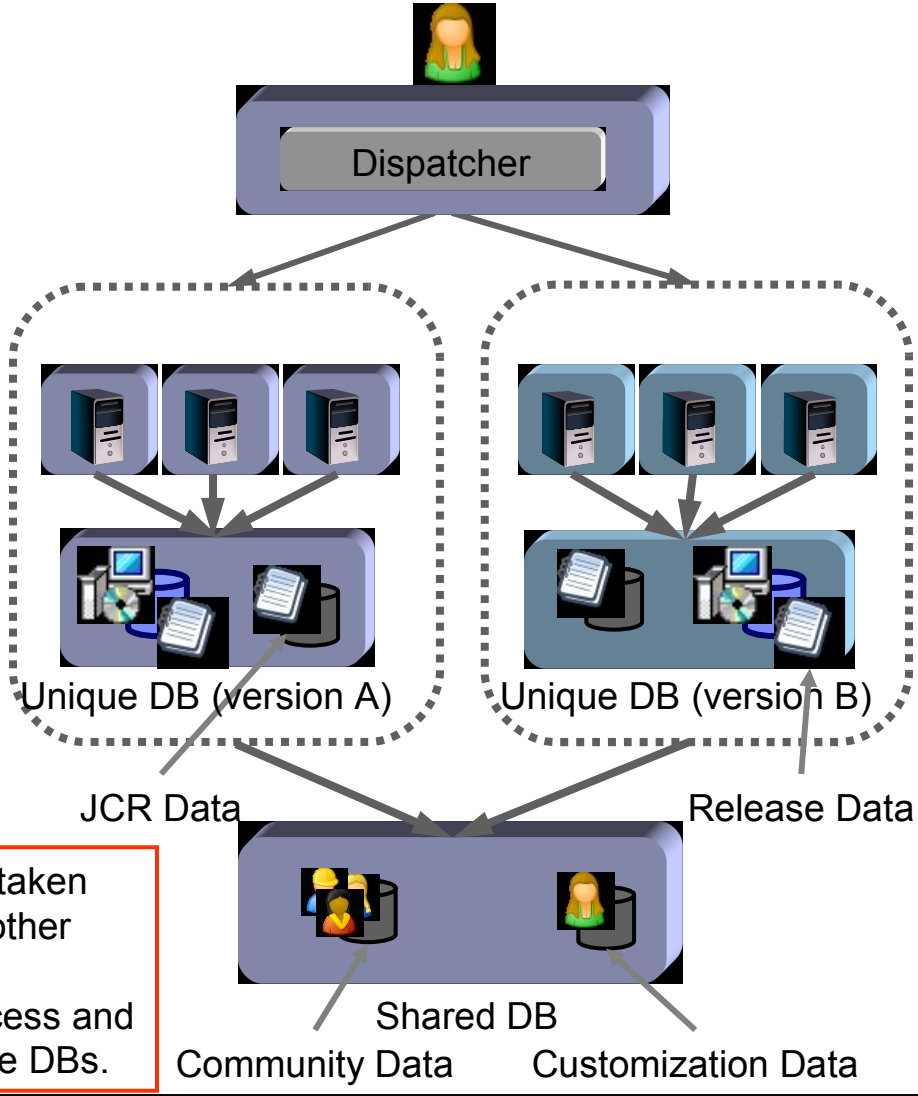
Site A and Site B concurrently serve user requests.
DR and continuous availability



Analysis for Availability – Database Sharing

Influence of the databases shared between clusters

Portal Solution Upgrades can be applied to one site at a time.
User customizations are preserved and can be cleand-up automatically when both sites are on the new solution release.
(**SLCheckerTool**)



Shared Databases can be taken offline for maintenance or other operations.
Portal Clusters can still access and serve Content from Release DBs.

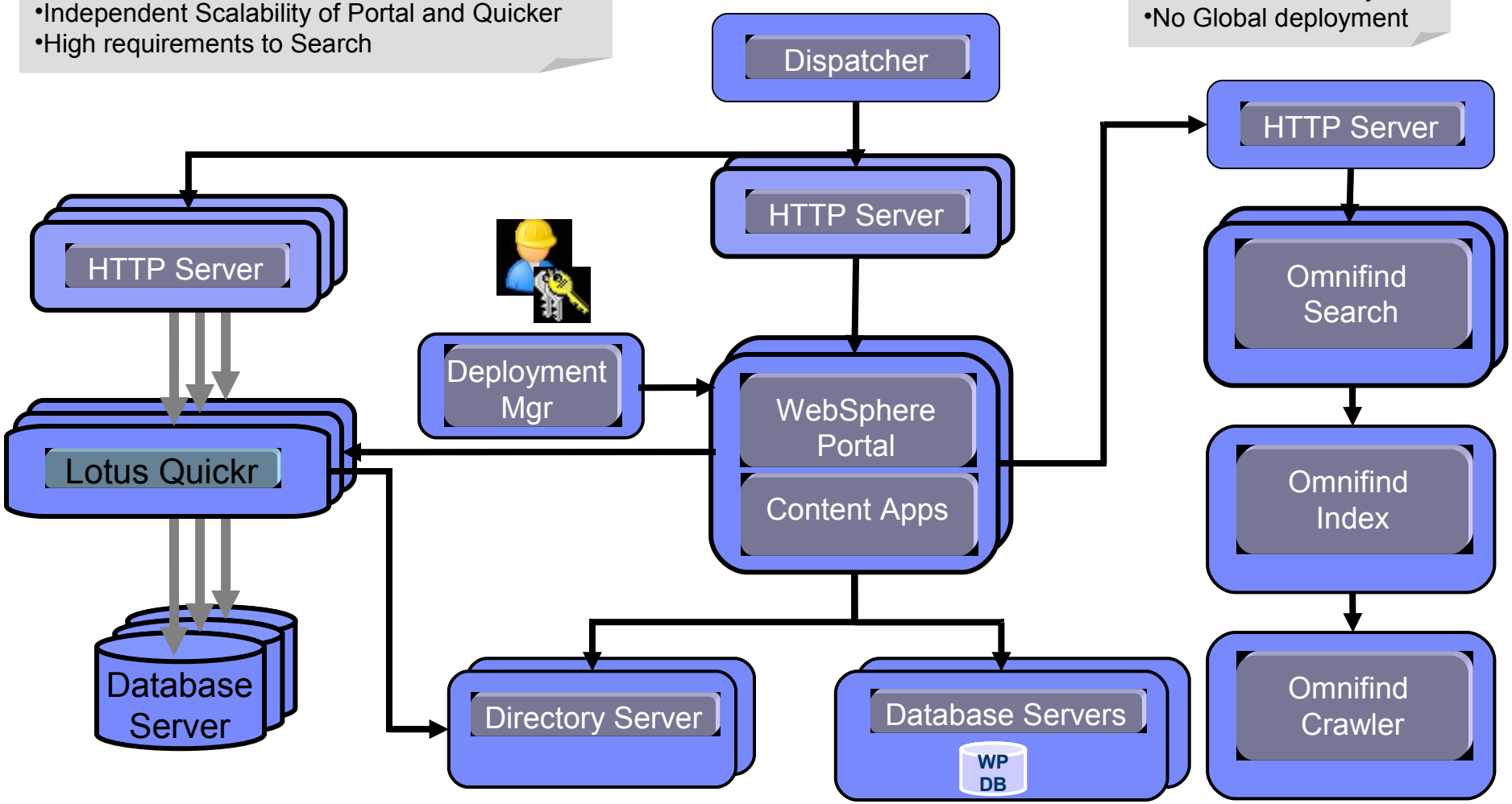
Modifications in Offline Mode
In this case users will be presented with the default release data configuration and will not be able to customize the portal. Admins are still able to modify shared resources.

Operation Characteristic
Maintenance of the shared DBs does not require downtime of the production system.

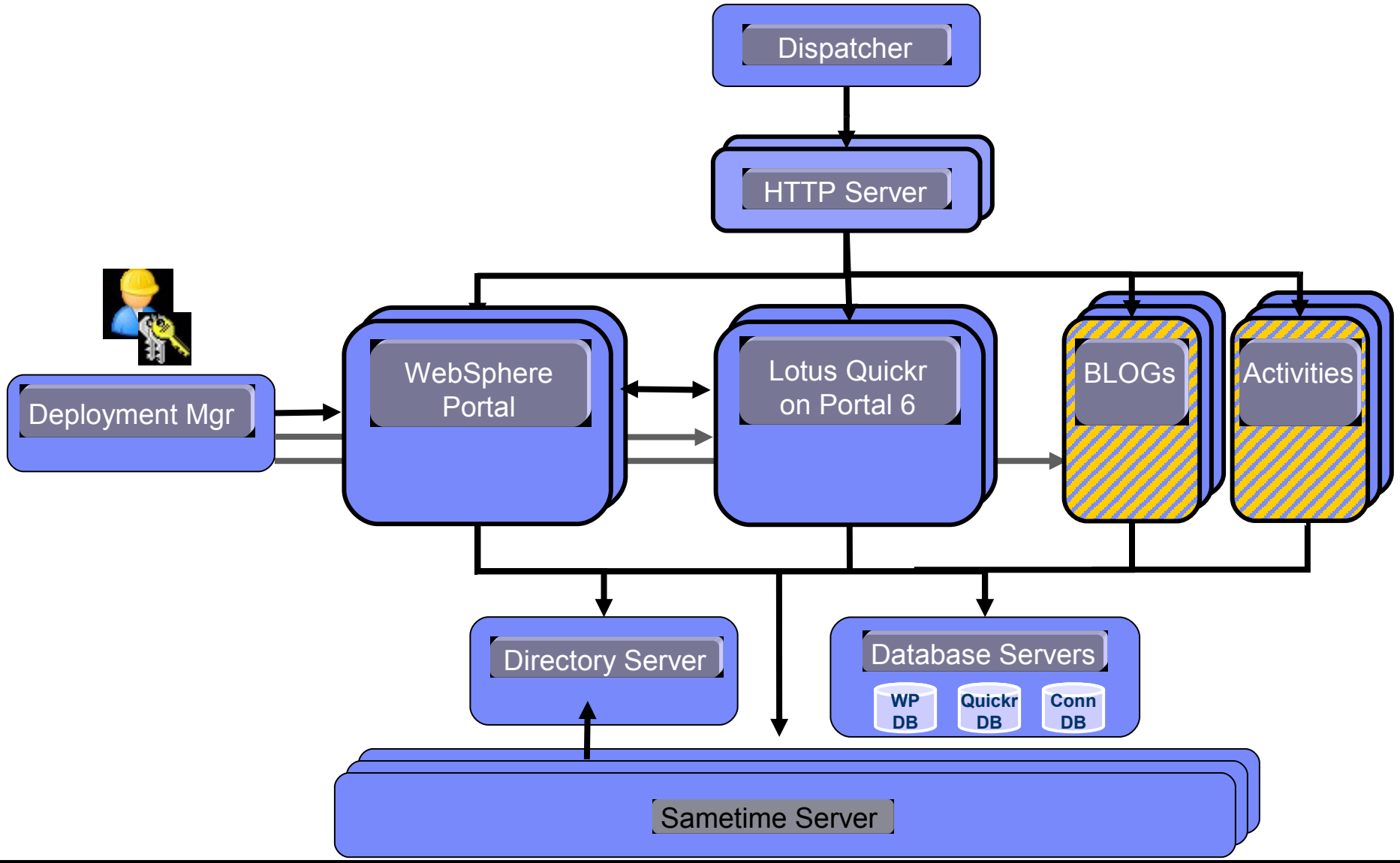
High scalable Content Accelerator – Quickr, WCM, Omnifind

- High availability
- Independent Scalability of Portal and Quicker
- High requirements to Search

- No 24x7
- Low content activity
- No Global deployment



High scalable Collaboration Accelerator – Functional Split



Recap on changes between Portal 6.0 and 6.1

Recap on some differences in Portal 6.0.x vs 6.1.x

Term	Version 6.0	Version 6.1
Member managementservice	WebSphere Member Manager	Virtual Member Manager
Database naming	Lookaside database	property extension database

Task	Version 6.0	Version 6.1
Configuring WebSpherePortal	Used scripts	Use tasks
Command	wpsconfig.bat|sh	ConfigEngine.bat|sh
Properties file names	wpconfig.properties wpconfig_dbdomain.properties wpconfig_dbtype.properties	wkplc.properties wkplc_comp.properties wkplc_dbtype.properties
Portal Scripting Interface Language	JACL	Jython introduced, JACL still supported but will be phased out

Portal Command Line Interface

Configuration Engine

- Also available as GUI
- Aka wpsconfig (in Portal 6.0)
- Unified framework for executing deployment and config tasks, eg:
  - Transfer data to a different database
  - Connect additional node to a database
  - Enable security / LDAP
  - Federating Portal Node
  - Deployment of Portlets
  - Maintenance tasks such as Portal URI change
- Properties files that is associated:
  - wkplc.properties - General Portal properties
  - wkplc_comp.properties - Portal configuration database detai
  - wkplc_dbtype.properties - Database configuration detail

Operation Tooling Overview


Developer

XYZ Artefact

Portal Application Archive


Administrator

Portal Script Jython

Config Console


Operator

Portal Config XML


Release Mgr.

Portal Config XML

ANT

Portal Deployment

wsadmin (JVM)

Portal Script

Console

Client JVM

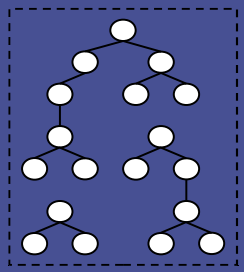
XMLAccess

ReleaseBuilder

SLChecker

Application Server

Portal Server



WSADMIN based
Integrated console
- Runtime configuration
- logging and tracing

SSL / HTTP(S)
- supported for XMLAccess
- *XMLAccess Only, or Web Access + XMLAccess*

Portal Scripting Interface

•Allows portal development teams to write scripts that are later executed by operation teams for deployment

•Can have the same functionality as the portal administration user interface

•Allows implementation of automated configuration management for various kinds of configuration changes

- Provide repeatability and avoid user errors

•The Portal Scripting Interface provides delegated administration in the same manner as the portal administrative user interface.

•Other benefits:

- This allows distributed portal administration
- Different development teams can work on related portal updates without interfering with each other's work. Likewise, different administrative teams can perform the tasks of developing a solution, and deploying and operating that solution for production without knowing the inner work of the solution.
- *Release Staging and integration* - via this interface, a new release can be developed and tested on a staging system and then be integrated into the production system while the system is running.
- Interactive portal administrative tasks do not require a Web browser and apply portal configuration updates in real time
- Based on WAS scripting interface, and provides its own online help as part of the scripting environment

Portal Scripting Interface

•Starting with WebSphere Portal Version 6.1 the Portal Scripting Interface is based on Jython syntax, and the JACL syntax will be phased out in the future. Eg:

- To issue help in jacl – *$Help help*
- To issue help in jython – *print Help.help()*

•Some operational info:

- Interactive mode: `wpscript.bat -port port_number -user user_id -password password`
- Script mode:
  jython - wpscript.sh -portport_number **-fscript_file_name.py**
  jacl - wpscript.sh -port port_number **-f script_file_name.jacl**
- Options:
  - -lang – jacl or jython
  - -conntype – SOAP, RMI or None (scripting bean is not active, for help purpose only)

Jython

•Jython is a general-purpose high-level programming language, is an implementation of the Python programming language written in Java

•Why Jython?

- Because we are inheriting the capability from wsadmin tool in WAS
- The use of JACL is deprecated in WAS 6.1 (hence Portal 6.1)
- Similarity to C/Java language syntax

•Some highlights of the language:

- It uses code indentation as block delimiters
- The character # starts a comment that extends to the end of the line
- By default, each line is interpreted as one statement, however, you can write multiple statements on one line by separating the statements with semicolons
- Jython is case-sensitive

JACL

•JACL is a Tcl interpreter that is written in Java, without strong typing. It is a procedural language with some object oriented concepts that are used by the scripting component.

•Some highlights of the language:

- The character # starts a comment that extends to the end of the line
- By default, each line is interpreted as one statement
- You can write multiple statements on one line by separating the statements with semicolons
- Statements can be nested by using square brackets [], which are interpreted like back quotes in most UNIX shells, the statement within is executed, and its result substituted in place of the bracketed statement before interpreting the surrounding statement
  - `outer statement [first inner] [second inner statement]`
- JACL is case-sensitive

Script Beans in Portal Scripting Interface

•The portal scripting component extends the wsadmin tool using “Script Beans”

•Portal objects, a particular page or an individual portlet are not represented by Jython or JACL objects

•Instead, a fixed number of Script Beans that provide access to specific areas of the portal data

•The list of available Script Beans are as below, leverage the help to get further understanding on how to use each beans.

- **Portal**
- **Content**
- **Layout**
- **Portlet**
- **Look (aka Theme/Skin)**
- **Access**
- **PacList**
- **Application**
- **ArchivedApplication**
- **ApplicationCategory**
- **TemplateCategory**
- **Publish**

Jython Sample

```
# get help - for the completely lost
Portal.help()

# get help on a particular bean
Portlet.help()
Access.help()

# get help on a method of a bean
Portal.help("login")
Layout.help("select")

# get help on an extended help topic of a bean
Content.help("search-criteria")
```

JACL Sample

```
# get help - for the completely lost
$Portal help

# get help on a particular bean
$Portlet help
$Access help

# get help on a method of a bean
$Portal help login
$Layout help select

# get help on an extended help topic of a bean
$Content help search-criteria
```

Other Concepts in Portal Scripting

•Portal object - Most portal objects are represented in by an object identifier string based on object ID of portal. Eg: *_6_00KJL57F9D02H456_A*

•Tree navigation - Content, Layout, and Portlet beans each represent a tree hierarchy; You select a node by its ID using the *select* command

•Search - All beans with tree navigation support *searching*

•Attributes - All beans use similar commands to query/modify attributes, some example of *Attributes*

- List valued attributes
- Locale specific attributes
- URL attributes
- Metadata attribuates

•Organisations and Hierarchy - For some beans, in particular the Content and Layout beans, the order of nodes is significant. *Move* can help reorder the node. Additionally, using *move/adopt*, you can move the whole subtree to another part of the tree

Site Management Extensions of Portal Scripting Interface

•Using site management extension bean of the Portal Scripting Interface, you can publish, promote, or demote portal content

- Similar to what the Site Management Portlets (GUI) offers
- Portal content can consist of:
  - Content nodes
  - Labels
  - Pages or partial hierarchies
  - Compositions
  - Links to both internal and external URLs
- Limitations:
  - You can only publish pages that have a unique name
  - You can go back only one version on a publish, promote, and demote scenario (For more, you need to create backup using XMLAccess)
  - Site management publishing does not support cross page wires

Example Code Snippets

- Publishing a page (JACL):
  - $Publish publishPage http://localhost:10040/wps/mycontenthandler wpadmin password http://remotehost:10040/wps/mycontenthandler wpadmin password testPage ibm.portal.Home null null true true
- Promote page (JACL):
  - $Publish promotePage http://remotehost:10040/wps/mycontenthandler wpadmin password com.ibm.portal.published_testPage 0 true true
- Demote page (JACL):
  - $Publish demotePage http://remotehost:10040/wps/mycontenthandler wpadmin password testPage true true

XML Configuration Interface (XML Access)

•It is an XML File that contains portal configuration (pages/portlets/access control etc.)

•Provides a batch processing interface for Portal configuration, eg:

- Deploy portlets
- Activate portlets
- Deploy configuration
- Create pages
- Create and populate virtual portals
- Export part or whole Portal config to an XML file, and recreate on another Portal
- Assign access control etc

•User must be a Portal administrator and password sent through HTTP or HTTPS

•Common things to be aware of:

- Each portal resource referred to by objectid – unique !!
- Generally, you'll be interested only in export of the release database domain (ExportRelease.xml)
- Majority of the times, the error will be available in *output.xml* file
- Problems related to xmlaccess could be varied – ranging from **duplicatekeyexception** (xmlaccess tried to create a resource and a resource with that key already exists in db) to couldnt parse xml

•XMLAccess is a very powerful administration and debug tool – everything you can do via portal administration, you can do via xmlaccess – And MORE !

Use of XMLAccess interface for Backup/Restore

•Use of the XML configuration interface for backing up or restoring ***complete portal configurations*** has following limitations:

- A complete XML export of a portal configuration is not sufficient to re-create the portal, you will need:
  - WAR files for your portlets
  - Any additional file resources, such as theme files if they are not part of the standard portal installation.
- The XML configuration interface is not designed to deal efficiently with large volumes of data
- For a backup and restore solution on a production server, you should rely on low-level database and file system backups.

XMLAccess – Security Concerns

- To use the XML configuration interface, you need to have:
  - *Manager role* on the virtual resource XML_ACCESS
  - *Security Administrator role* on the virtual resource PORTAL
- This implies that you must be a Portal Administrator role who can perform any action (we will discuss this further in Security section)
- Hence there are no further access control checks that could restrict your actions when you use the XML configuration interface
  - you may view all resources in the portal and you may update and delete all resources.
- To run the XMLAccess, you must authenticate by specifying portal user ID and password
  - if you use an HTTP connection, the user and password are sent to the server unencrypted – which means your password can be compromised
  - Hence, configure SSL and use a secure HTTPS connection to connect to the XML configuration interface to avoid security breach

Overall structure of the XML input and output

•Requests to and responses from the XML configuration interface use the same XML format

•An export request generates an XML response that contains all the configuration data required to re-create the exported configuration part

- Hence you can export a portal configuration, save the XML output file andsend it to another portal to re-create the same configuration there

•3 types of requests that can be sent to the XML configuration interface:

- *Export requests* – It triggers the export of complete or partial portal configurations into XML response file, and it does not modify the configuration of the existing portal
- *Update requests* – The request modifies the configuration of the existing portal according to the values found in the input XML script
- *Export-orphaned-data requests* – Such request exports the complete portal configuration into XML, including orphaned data, and results in a response file

Difference – Portal Scripting vs XMLAccess

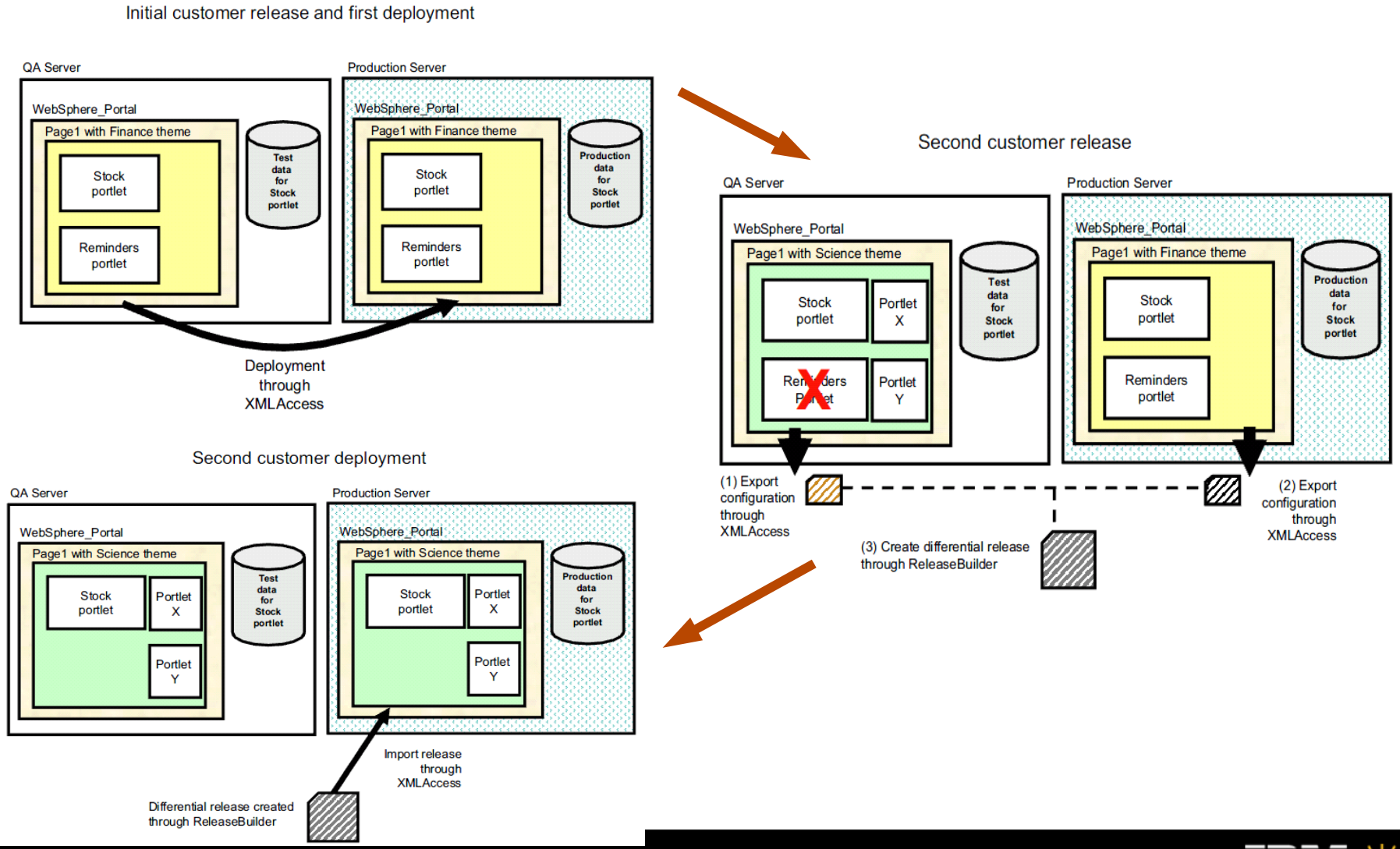
- Portal Scripting vs. XMLAccess
  - Most importantly, Scripts can access the Portal as any userid, not just as a "super administrator“
    - Provides delegated administration
    - XMLAccess interface does not allow for easy separation of distributed portal administration (as you need to use “Portal Administrator” user)
  - Scripts offer better “fine-tuned” administration
    - Same functionality as the portal administration user interface
  - Scripts provide easier automation of repetitive administrative tasks

Portal Deployment

ReleaseBuilder Tool

- ReleaseBuilder enables management of release configurations independent of user configurations
- The Release Builder tool is used to compare two XML Access files, determine their differences and produce a new script that can be used to synchronize the two environments
- Summary of its capabilities:
  - Compare two XMLAccess files
  - Determine the differences
  - Produce a new script that can be used to synchronize the two environments
- Release Builder is valuable to identify portal artifacts that are deployed to the production environment but are not longer in the stage environment
- It still leverage XMLAccess to process the output xml file produced that prescribe the differences between Rev 1 and Rev 2

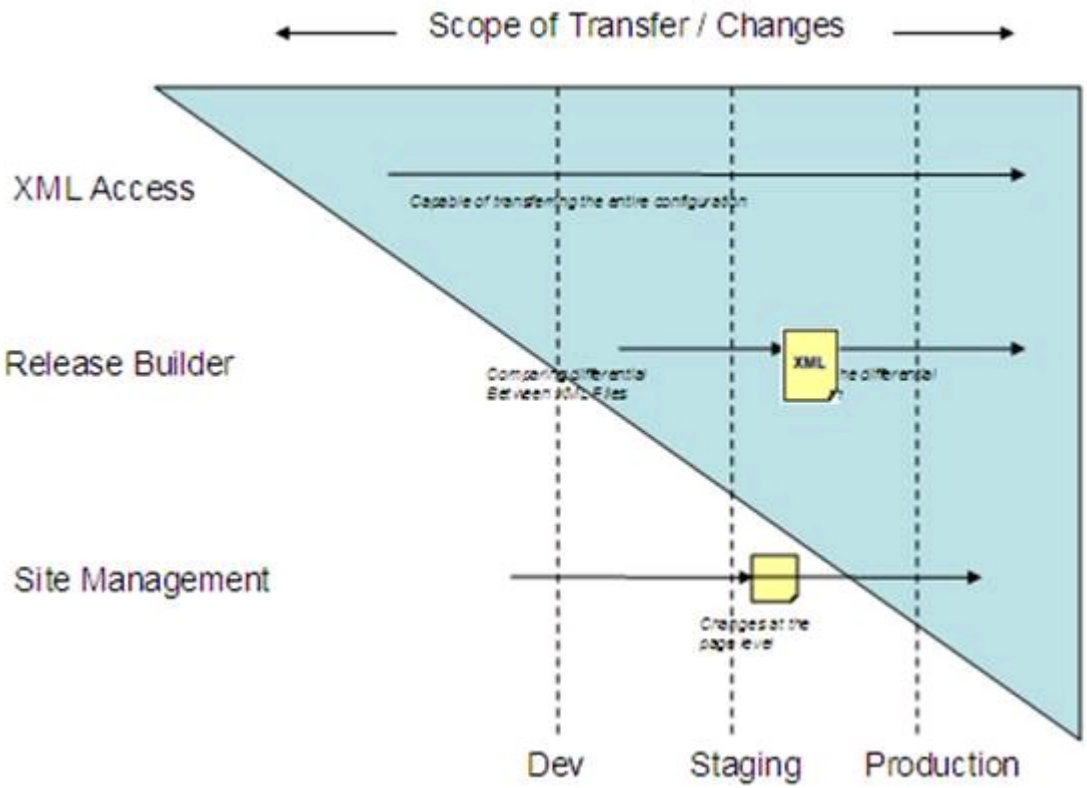
A typical ReleaseBuilder Procedure (Staging to Production)



Deployment Tools – When to use which?

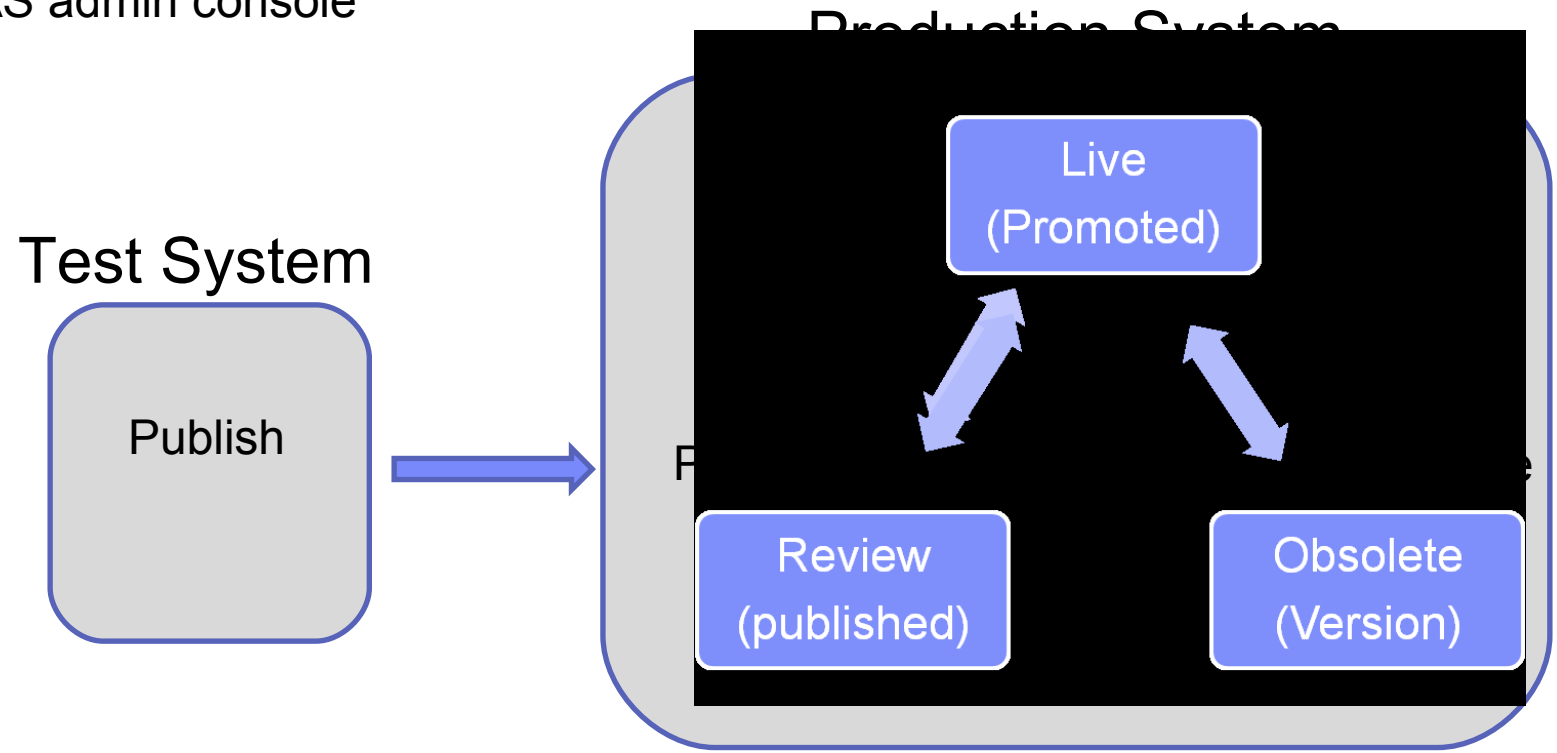
- For a complete transfer - use XMLAccess
  - e.g. An initial transfer from development to staging and to production, because we don't need the *differential option* that Release Builder provides
- Once you have already done the first complete transfer, you would want to transfer only what is *new, deleted, changed;* this is when we use ReleaseBuilder.
  - The differential option would create a XML with the recently created, removed, changed items.
- Finally, new Site Management Publish portlet in Portal 6.1 allows you to transfer a *specific page* or *page hierarchy*
  - Or via Site Management Bean in Portal Scripting Interface

Scope of Transfer / Changes



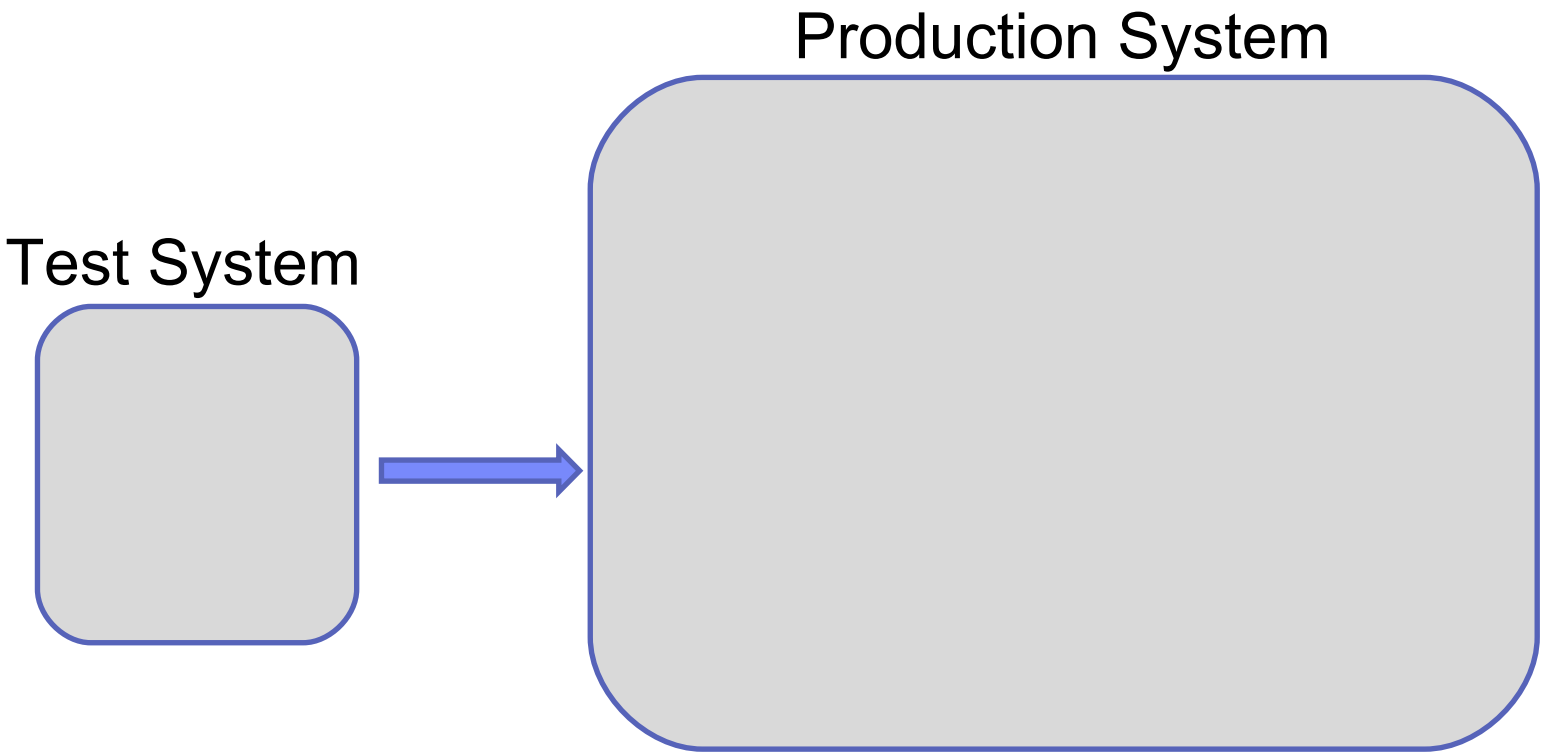
Deployment

- You can deploy from test to production servers using your choice of *command line* or *GUI tools*
- Site Management Publish GUI (New in V6.1)
  - You can also prevent administrative users from publishing, promoting or demoting pages using Site Management by disabling HTTPBasicAuthTAI in WAS admin console



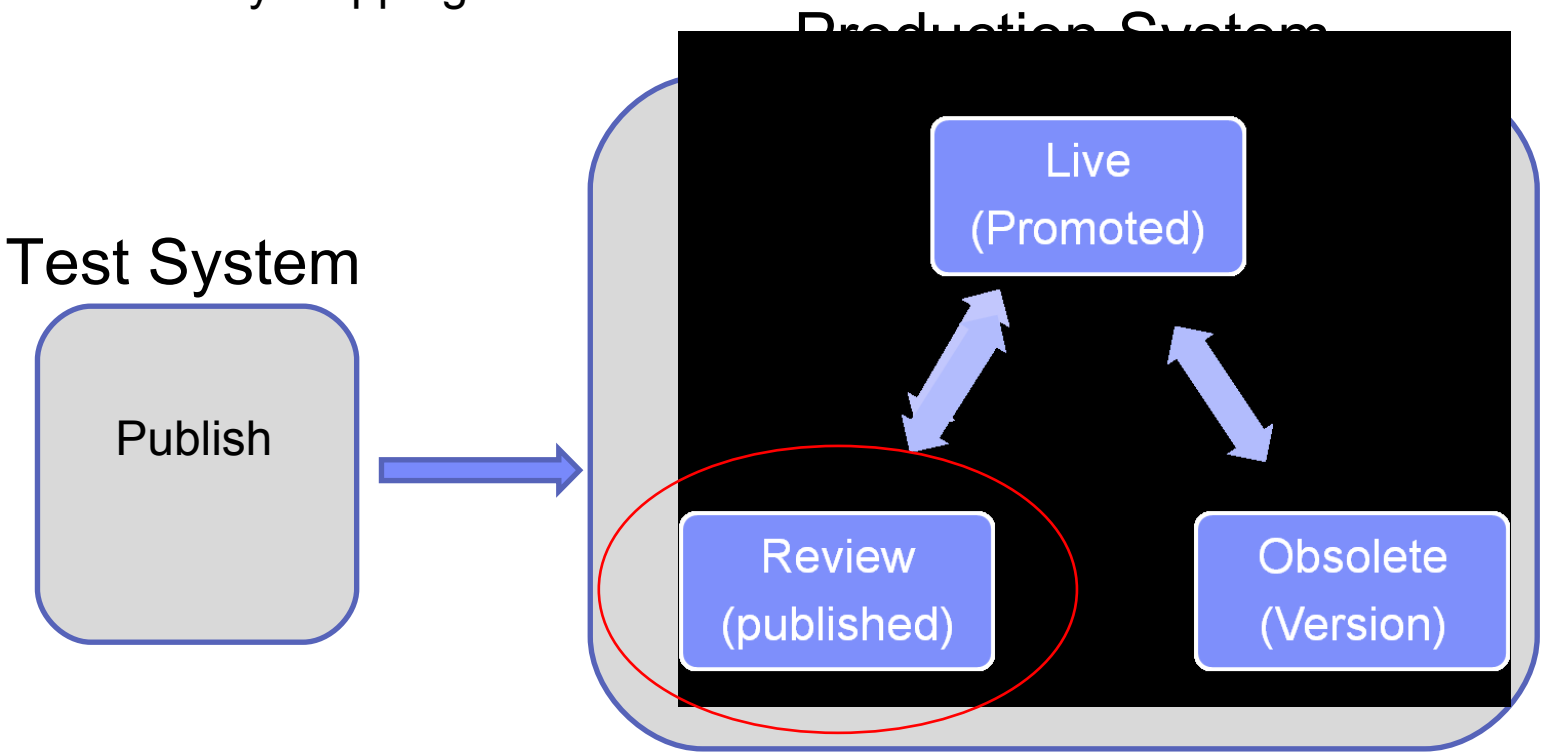
Site Management Publish Example

- A large retail furniture company has a test portal and a production portal.



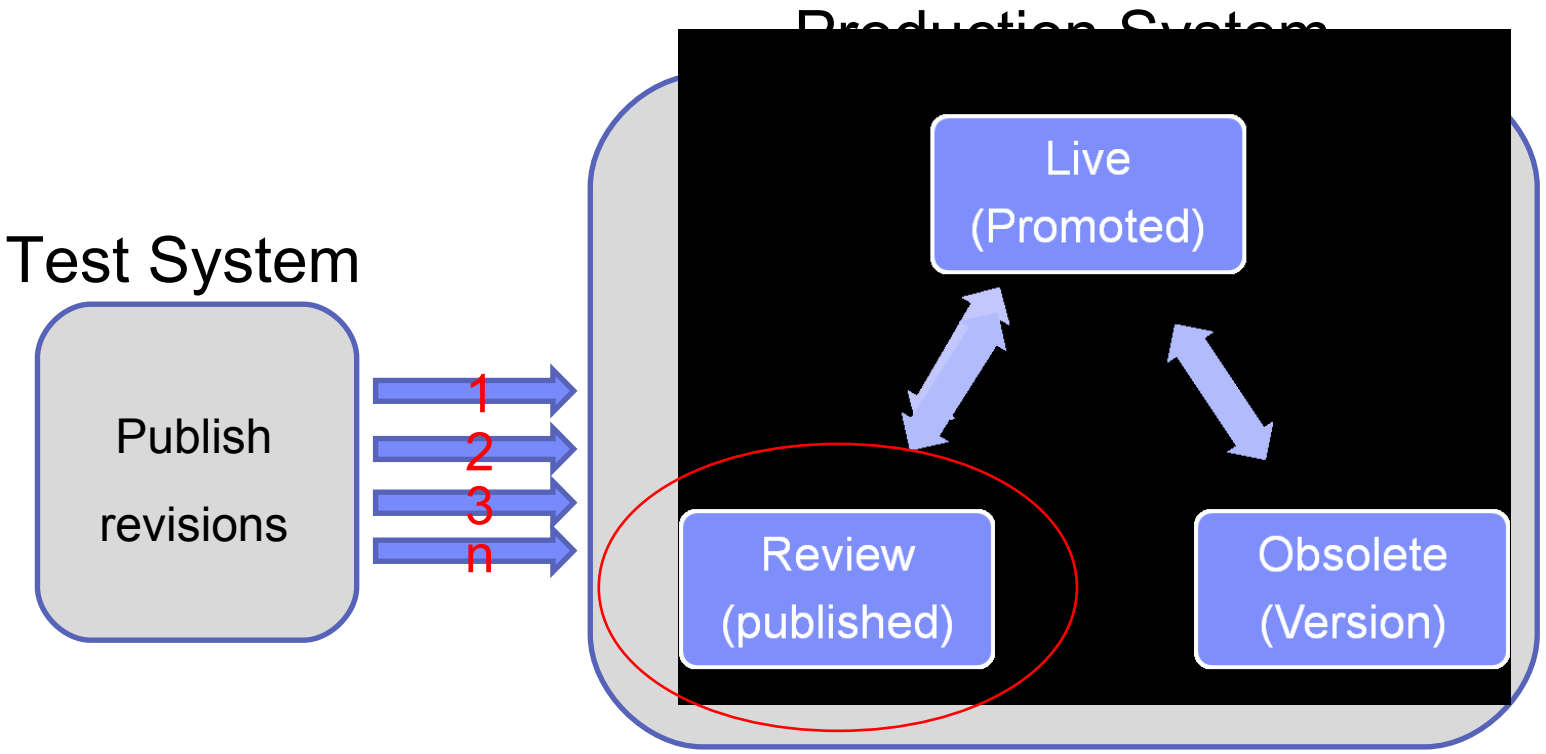
Site Management Publish Example

- The portal developer creates a page for a special sofa promotion on the test portal.
- The portal administrator publishes it to the production portal. As the page is published but not yet promoted, it can be reviewed by a limited set of users, including the marketing manager and the head of sales.
- Uses visibility mapping rules!



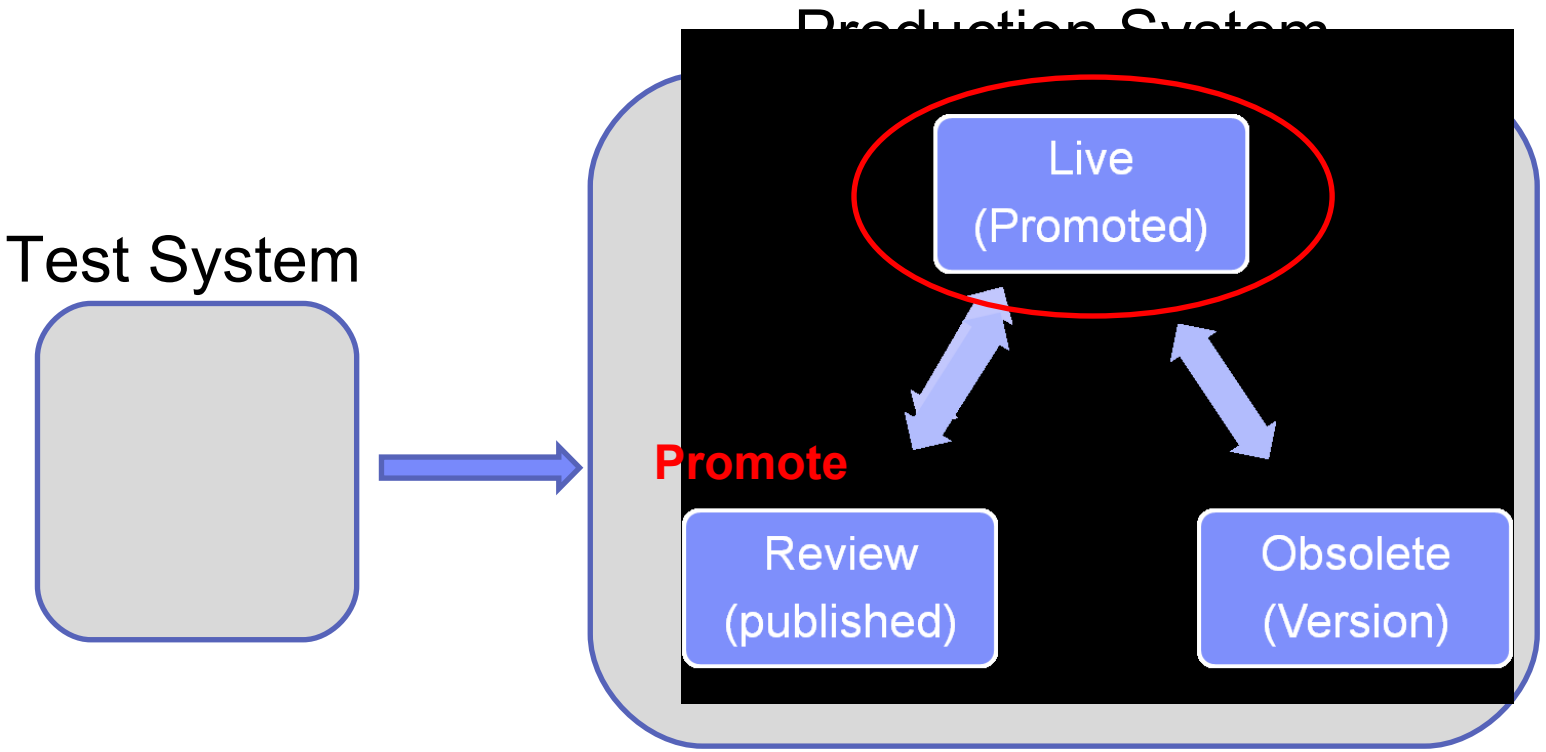
Site Management Publish Example

- When these reviewers look at the page, they request revisions to some of the page content. The developer can modify the page on the test server and the administrator can republish it to the production portal as many times as necessary.



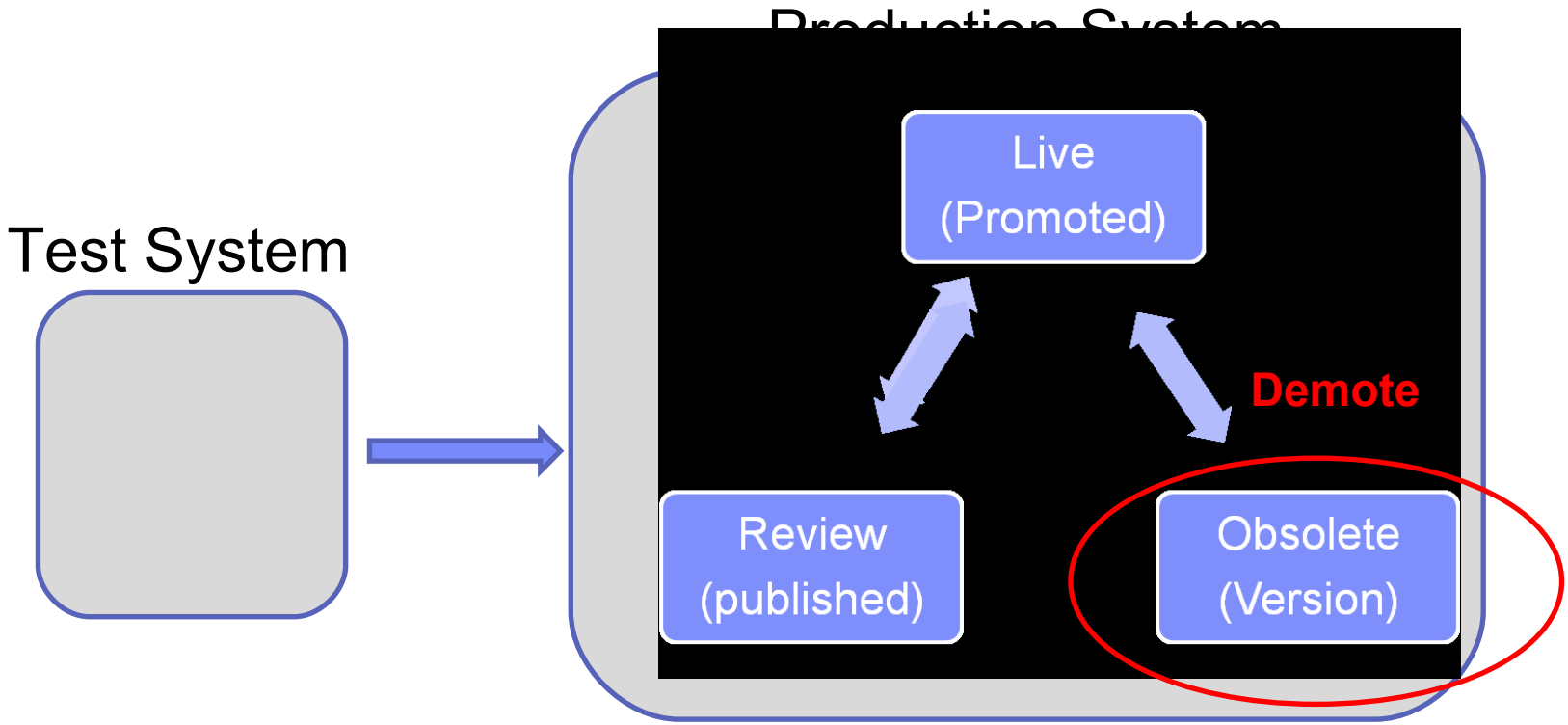
Site Management Publish Example

- After the reviewers approve the page, the administrator promotes the page to make it visible to all the furniture company sales staff who access the production server.

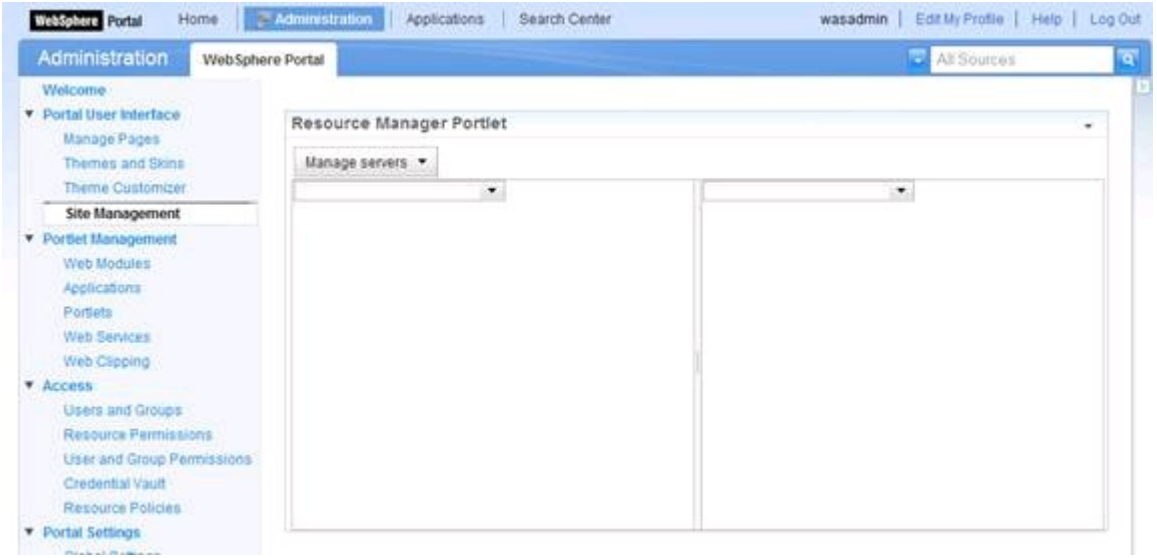


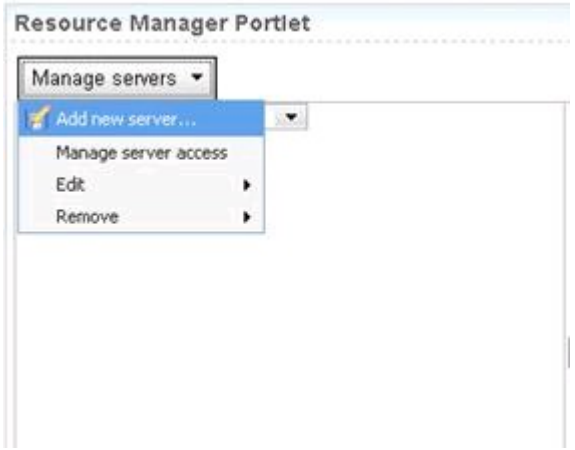
Site Management Publish Example

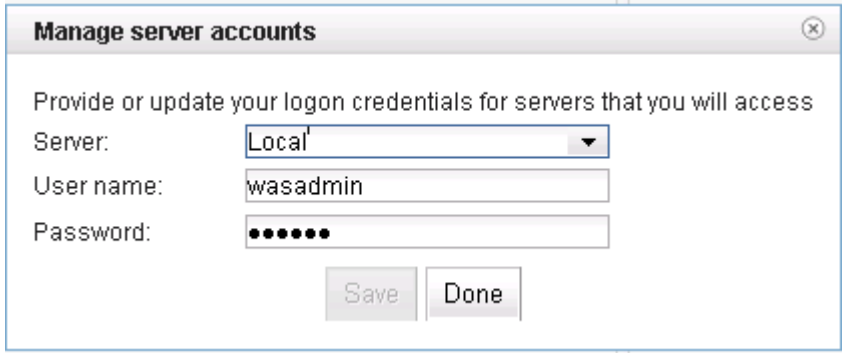
- When the promotion period is over, the administrator demotes the page on the production portal. Users can then no longer view the page.



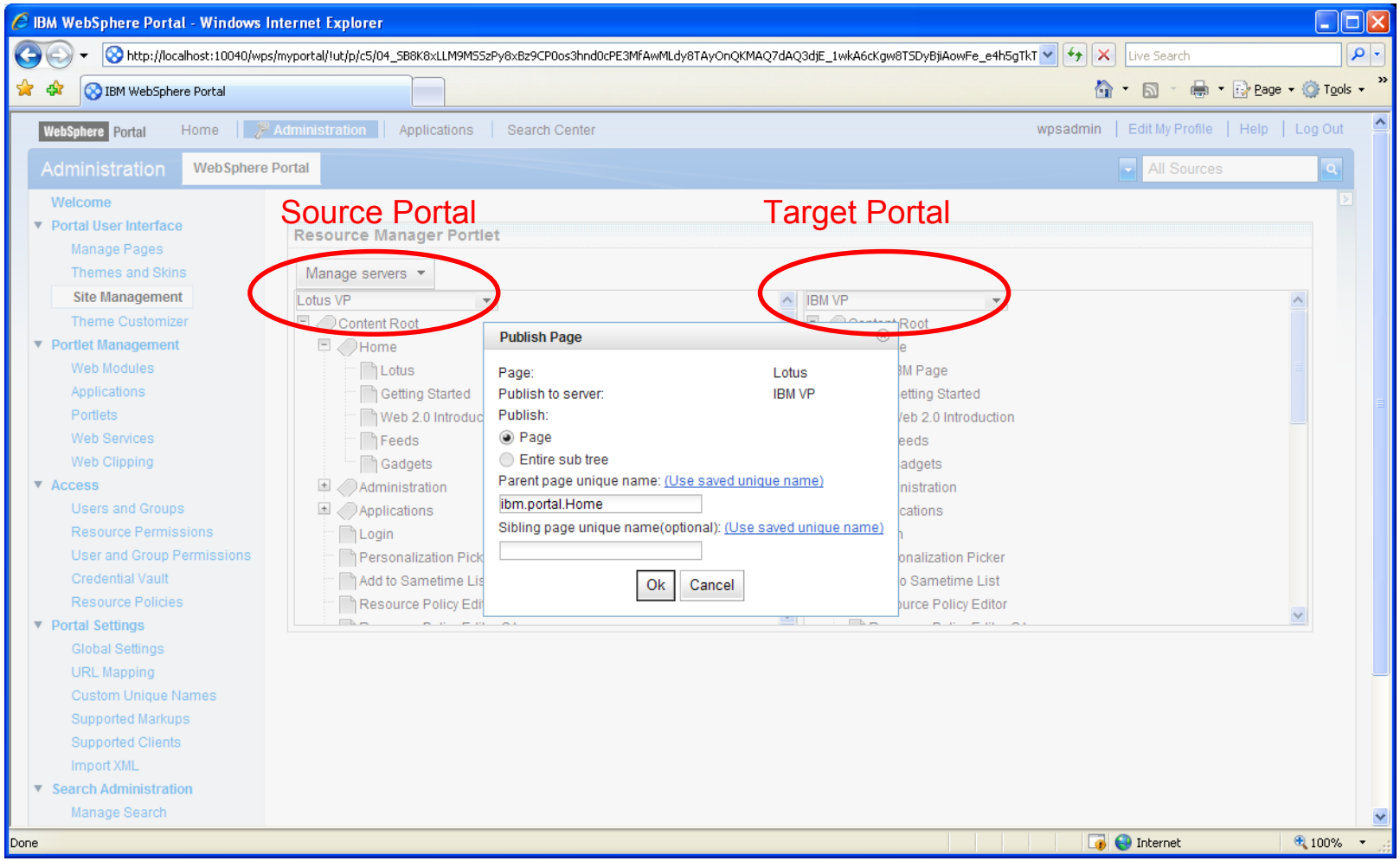
Site Management Publish GUI



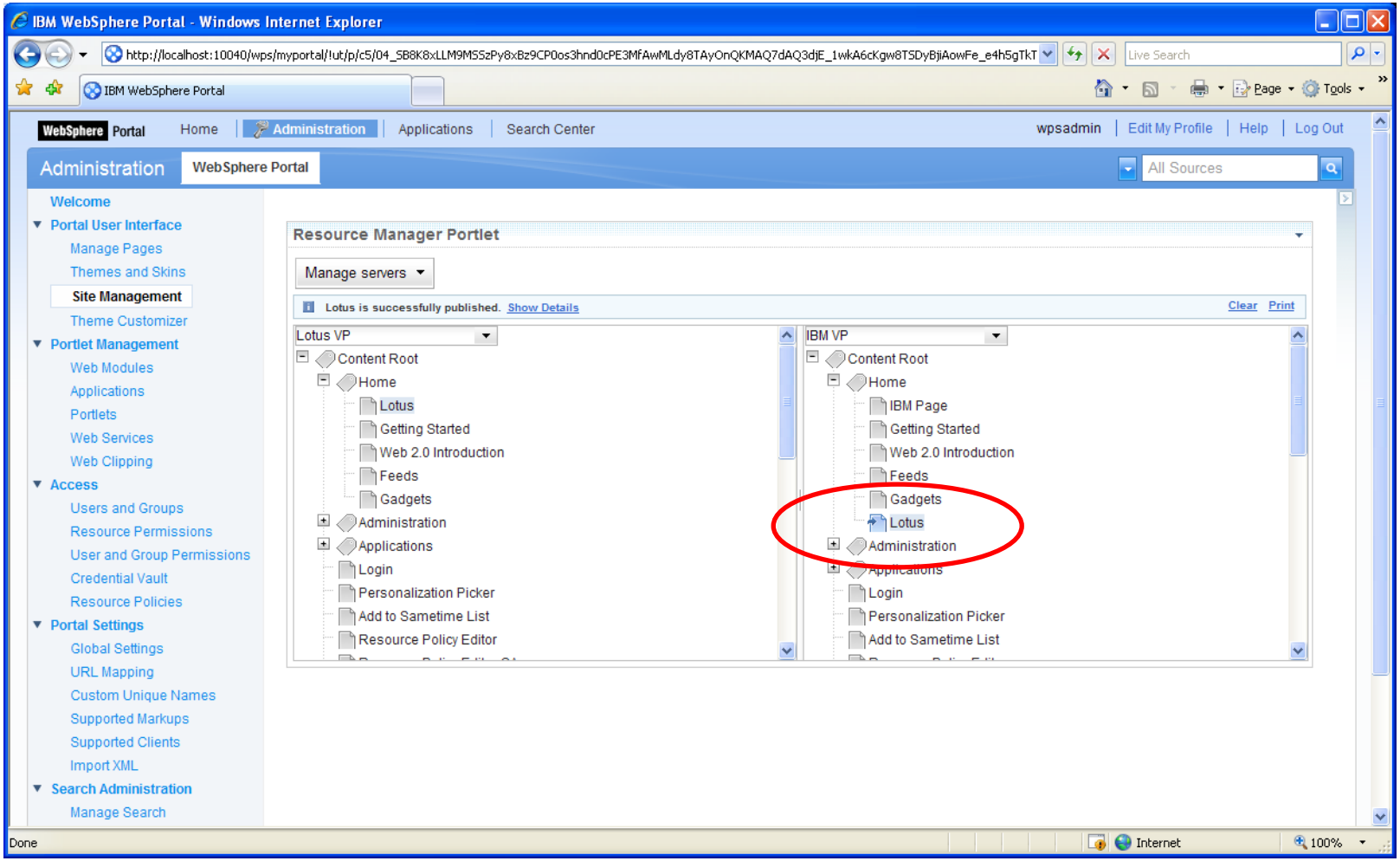




Site Management Publish GUI

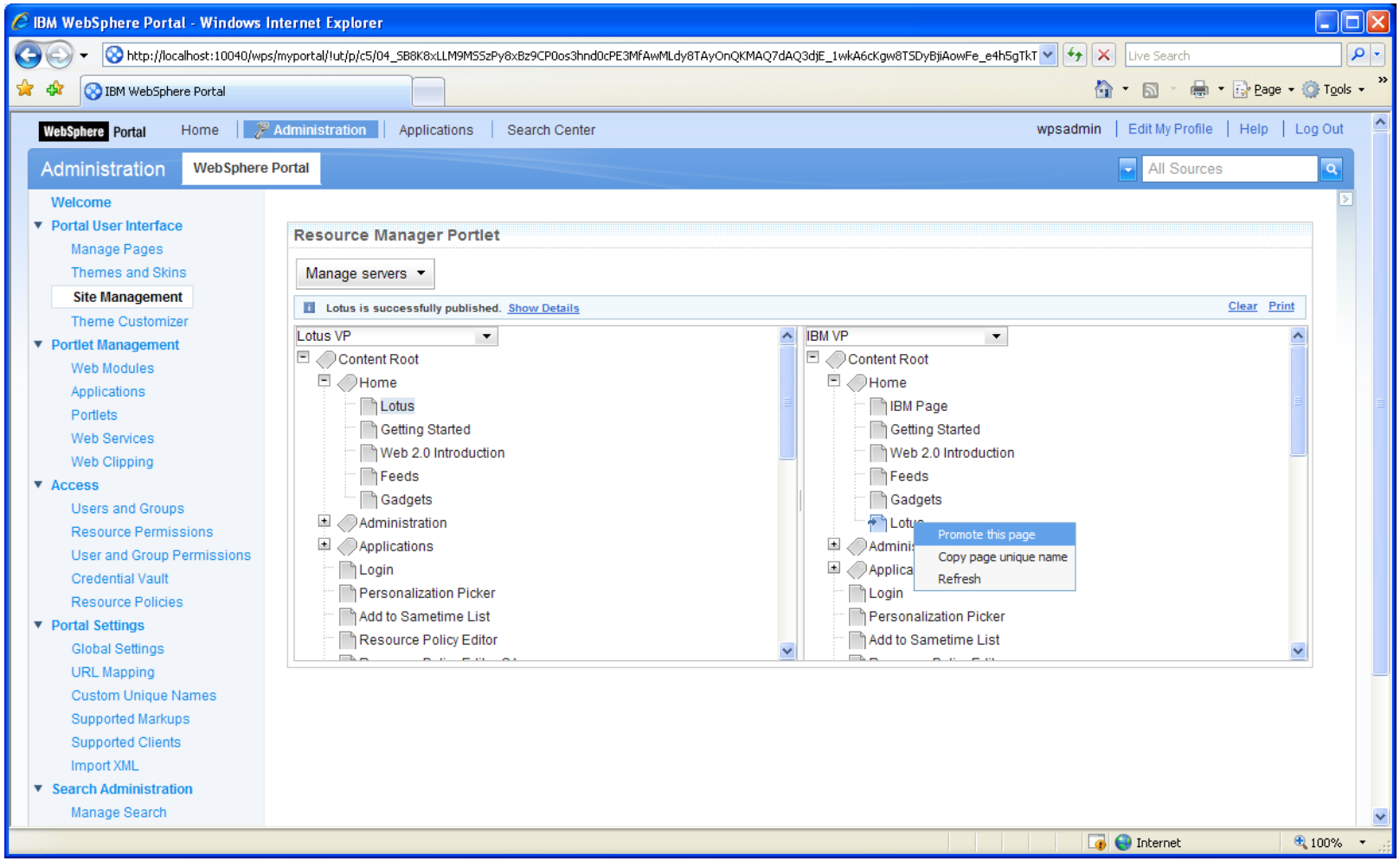


Site Management Publish GUI

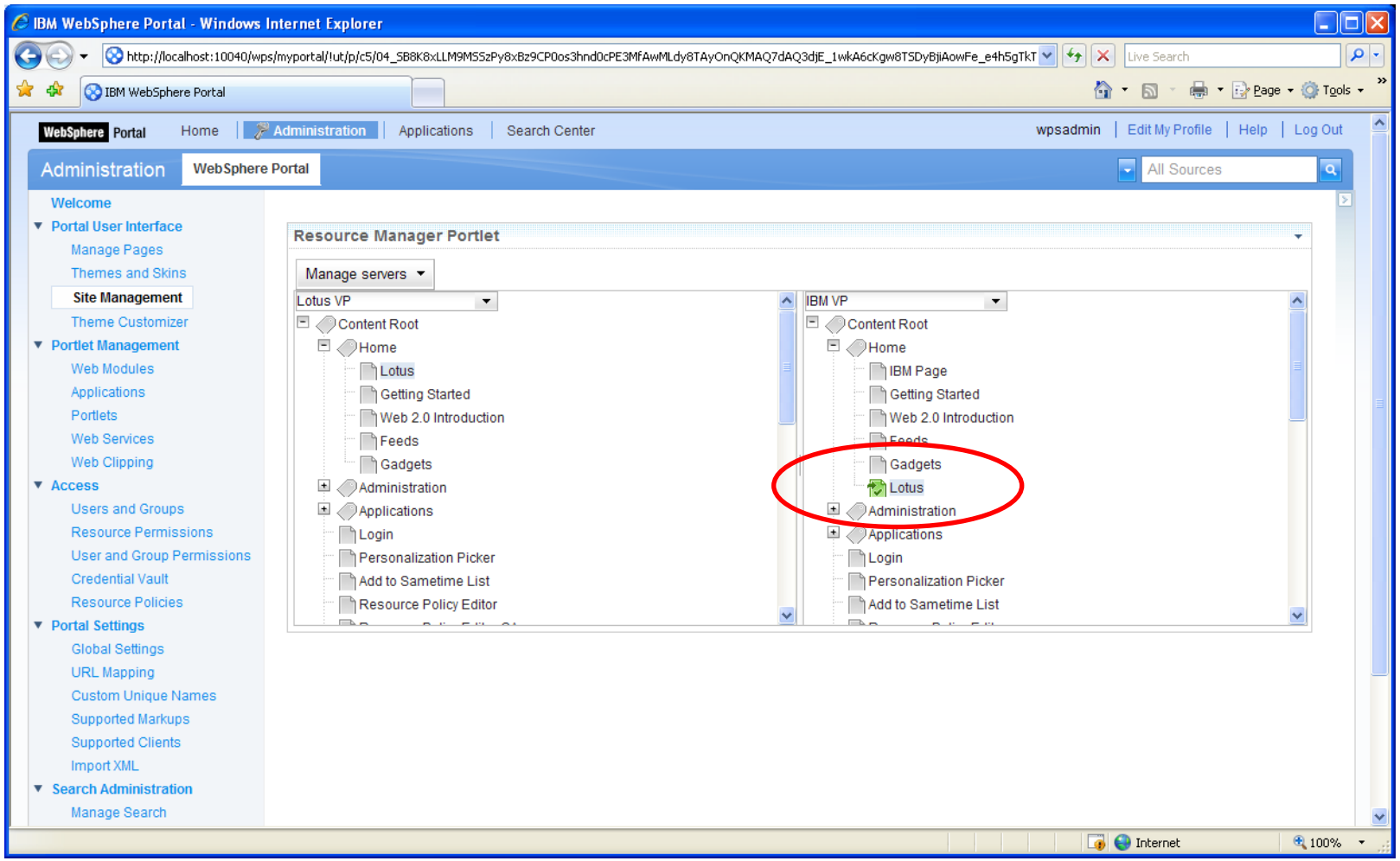


New page only visible to selected reviewers. Accomplished through Visibility Rule Mappings.

Site Management Publish GUI



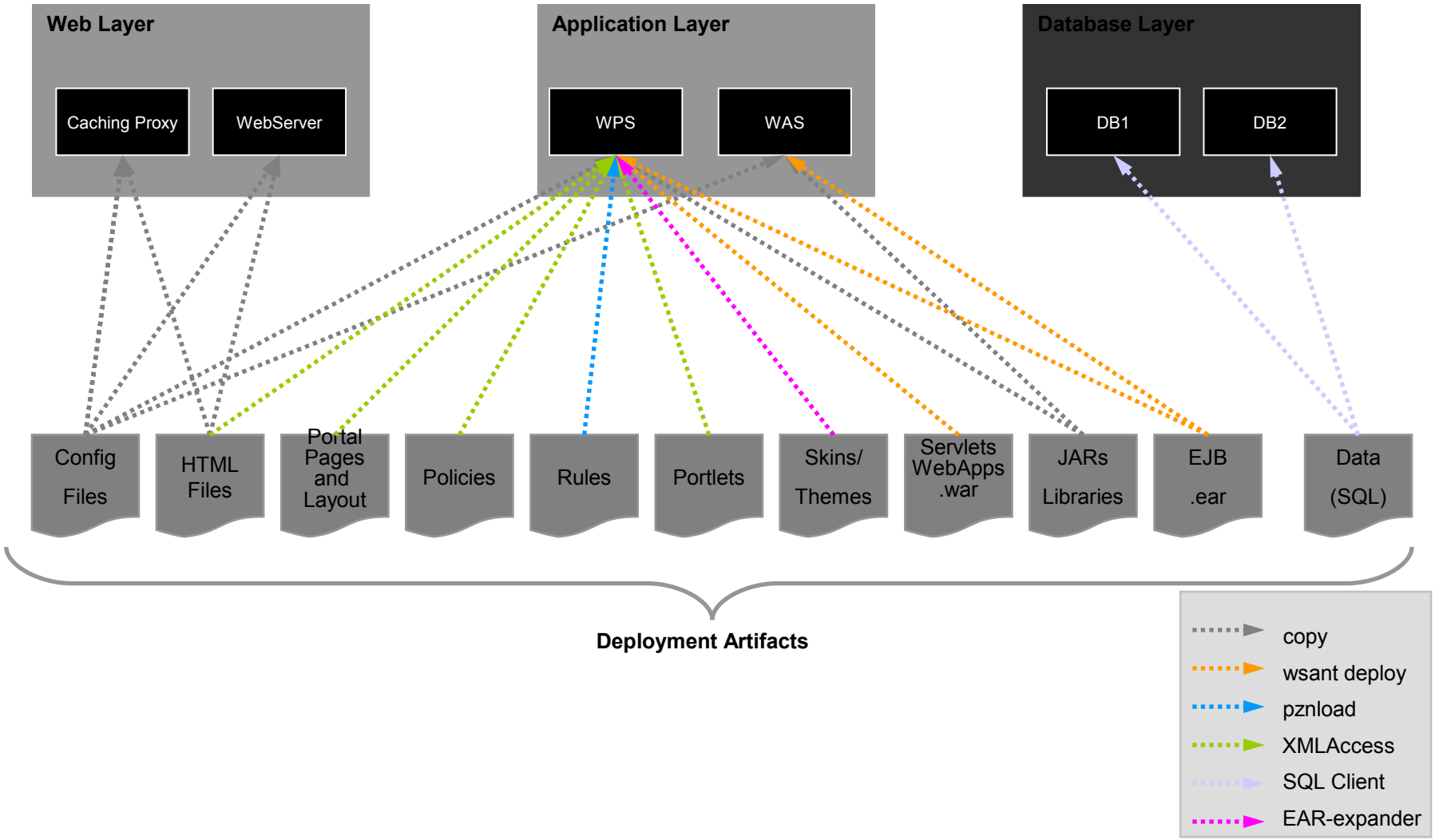
Site Management Publish GUI



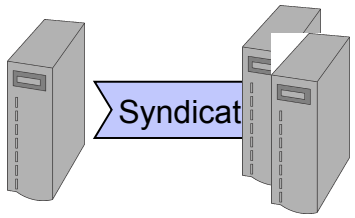
Promoted Sign - After the page is promoted, this sign will be shown next to the page

Other Portal Deployment Concerns

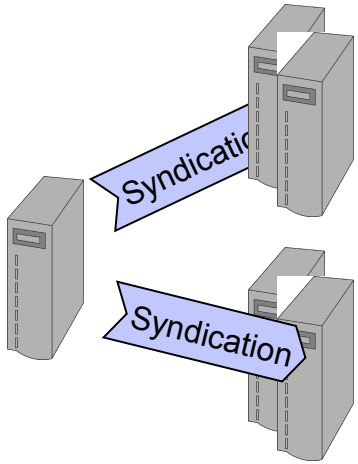
New Applications: Set of Artifacts you need to worry...

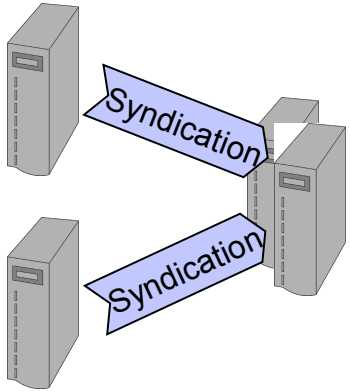


Content Publication – WCM Syndication

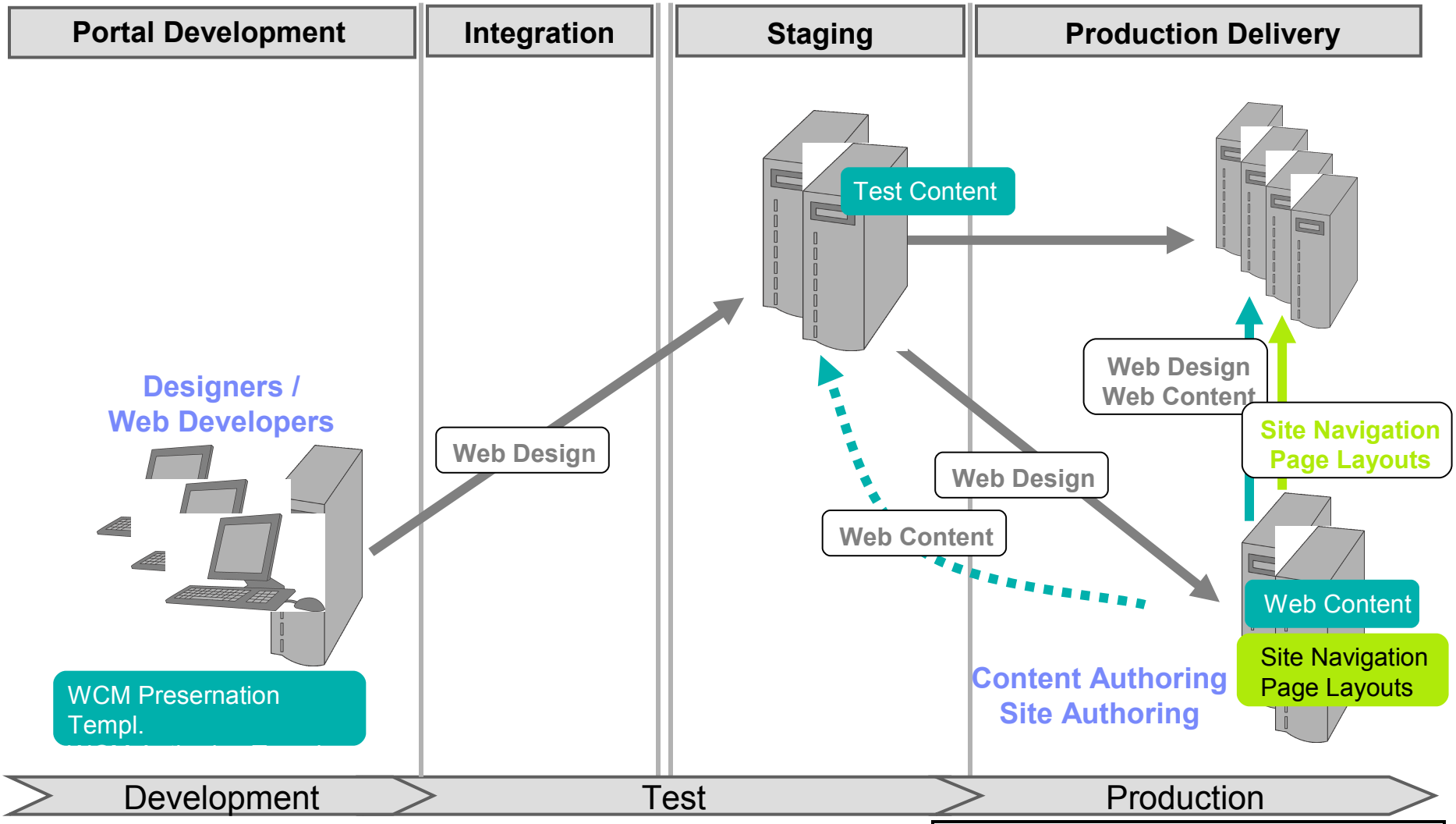


- **Configure and monitor syndication in the Portal Content Syndication Portlet.**
  - Typically used for automatic syndication (30sec - 2h)
- **Syndication allows multiple content architectures.**
  - one source to one target (1-1)
    (e.g. remote rendering or local rendering for single cluster)
  - one source to multiple targets (1-m)
    (e.g. HA Portals with local rendering)
  - multiple sources are supported as well (m-m)
    (e.g. decentralized content authoring)

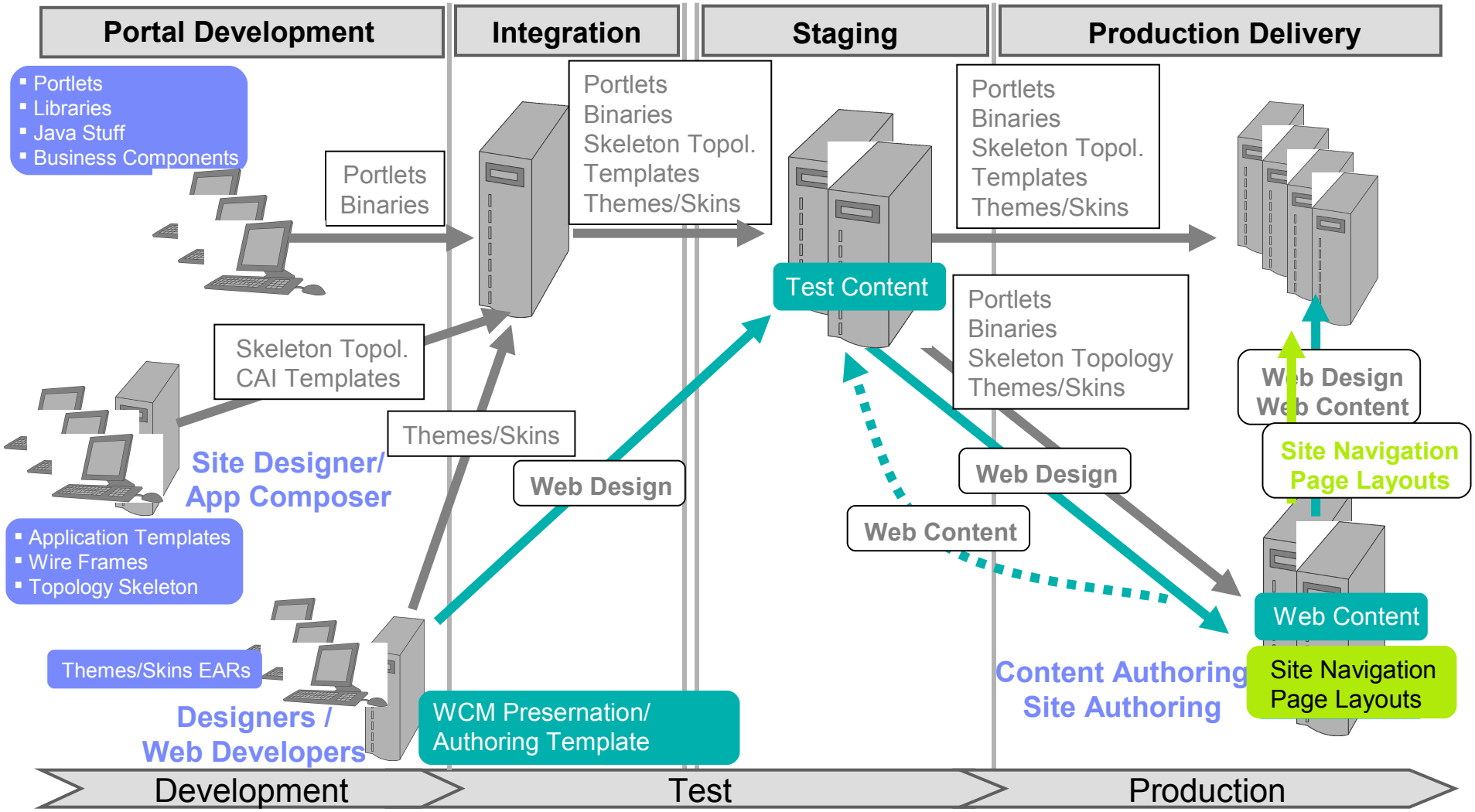




Portal 6.1 Data Management Blueprint – with Site Authoring

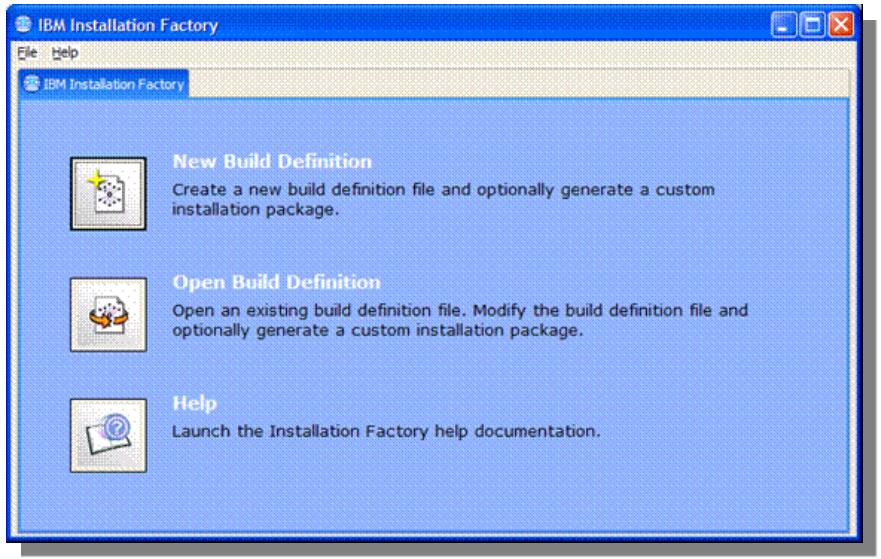


Portal 6.1 Data Management Blueprint



Portal Cloning – "Freeze dried" portals (An alternative approach)

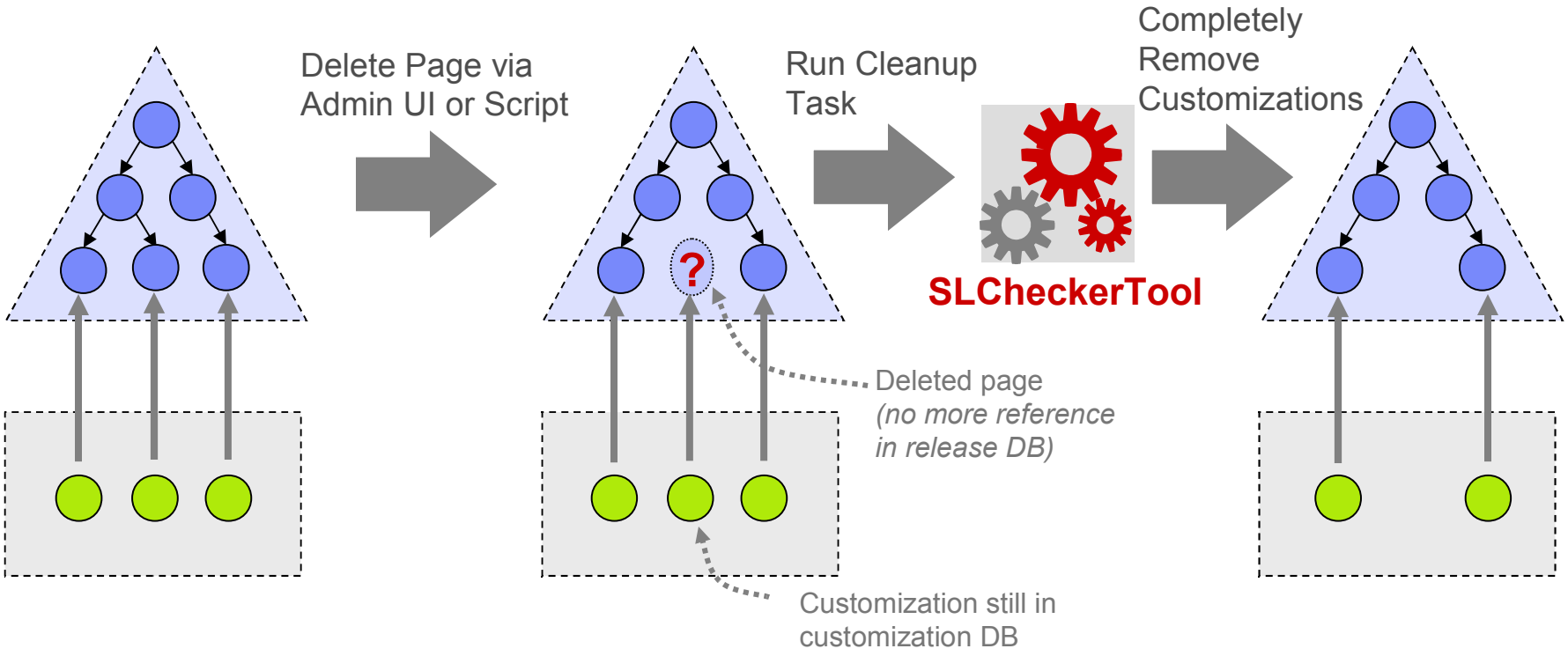
- Incur the cost of installation, full configuration and customization **once**
- Capture result and reuse as a template for additional instances
- Can be used as a basis for
  - new development environments, or
  - new production nodes
- Create an installation archive per environment
- Supports real and virtual hardware
- New instances created in 40 minutes, ready to go
- Leverages WAS Install Factory technology



Portal 6.0 - http://www-128.ibm.com/developerworks/websphere/library/techarticles/0704_lamb/0704_lamb.html
Portal 6.1 - http://www.ibm.com/developerworks/websphere/library/techarticles/0902_lamb/0902_lamb.html

Data Cleanup and Weak Reference Validation

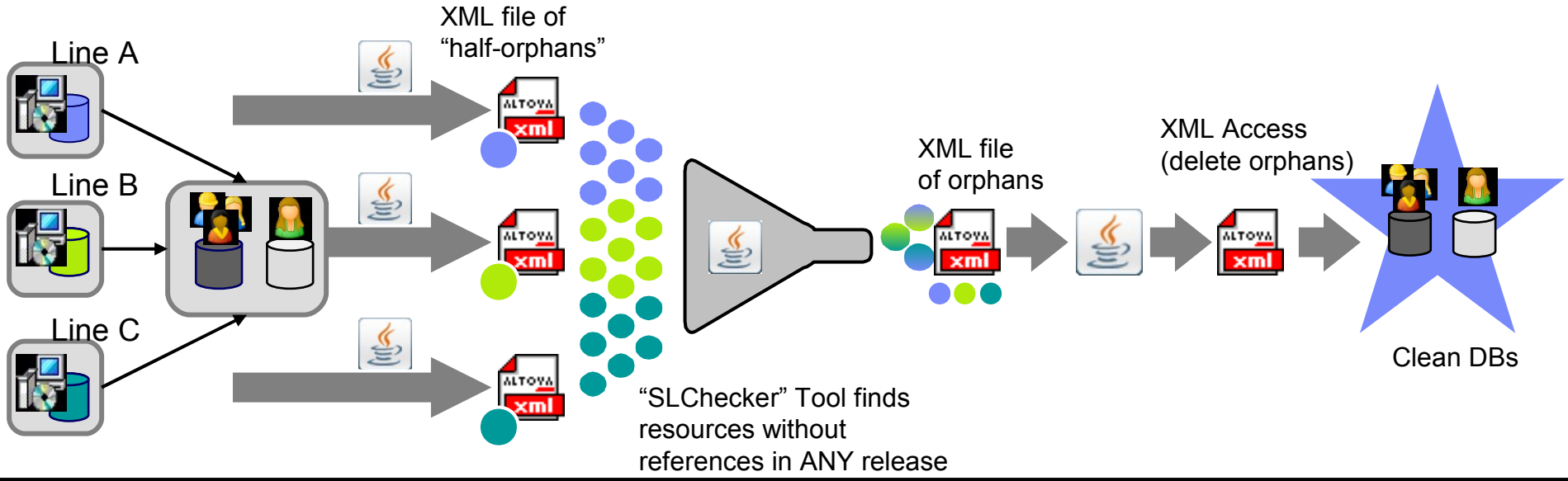
- Support multiple solution releases to work against a single customization domain.
- User customizations are kept until administrator decides to delete them.
  - No automatic cleanup of weakly references during administration
- Start a cleanup task to delete such references explicitly when needed.



Cleanup Wizard (SLChecker-Tool)

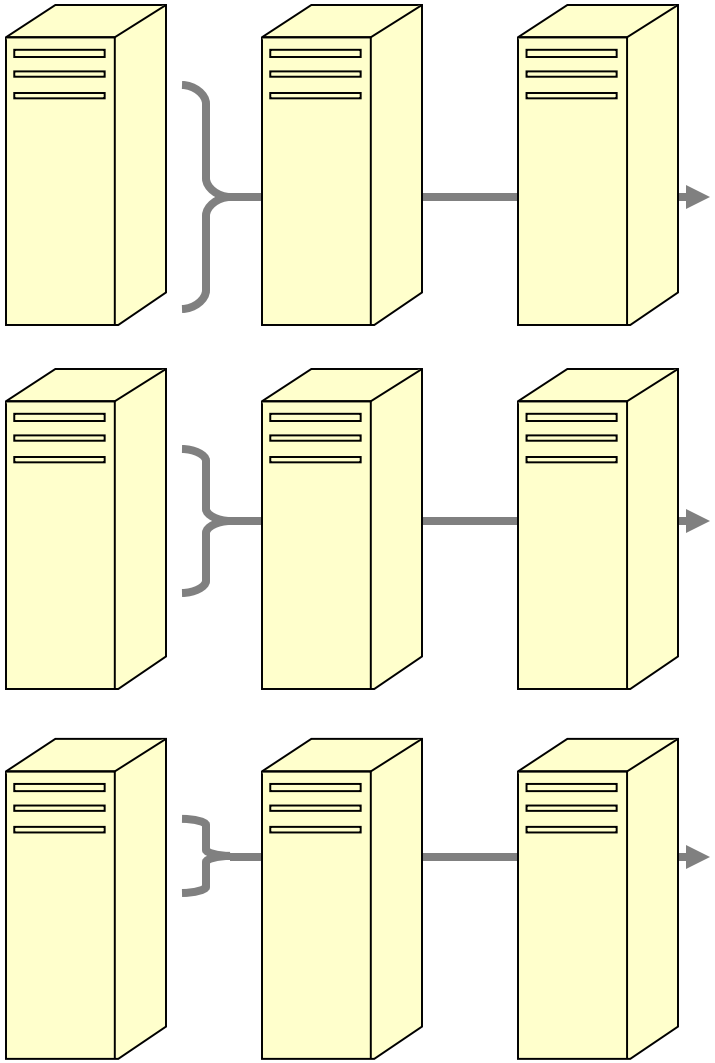
- Resources in the customization domain are not deleted with their referenced resources in release or community domain (Possibly still needed in backup scenario or by other production line).
- Orphaned entries accumulate in the customization domain.
- Remove orphaned entries → SLChecker

- Three steps to cleanup the customization domain
  1. Export Data (via XML-Access) from each production line that uses the customization domain to be cleaned.
  2. Run java-based Tool to extract the orphaned entries and generate .xml file for XML-Access
  3. Remove the orphaned entries using XML-Access



Virtualisation

- Operating system level virtualisation
  - VMware, XEN, etc
  - Normalises hardware
- Software level virtualisation
  - Portal cloning
  - "Normalises" the operating system
- Runtime virtualisation
  - Virtual Portal
  - Portal Profiles
  - WebSphere Virtual Enterprise
  - Normalises Portal



VMware Support

- Using VMware to support a continuous WebSphere Portal Beta
  - Started with WP 6.1
  - Will continue with WP 6.1.0.x/WAS 7.0 and WP 7.0
  - Novell SuSE SLES 10 guest OS
    - Licensed for beta use
- WebSphere Portal has a limited support statement for VMware (here)
  - Basically support VMware, but reserve the right to request recreation of problems outside VMware if it is suspected
- The Portal product team has more experience now with VMware
  - Have not seen any functional issues
  - Issues with performance and capacity – avoid swapping at any case
  - Observations will be shared once solidified through PortalZone or Portal Support
    - http://www.ibm.com/developerworks/websphere/zones/portal/
    - http://www.ibm.com/software/genservers/portal/support/

Overview of Portal Cluster Setup

Broadly – Steps to build a Portal Cell

- This is a multiple steps process

1) Creating a Deployment Manager
2) Install WebSphere Application Server Network Deploy (WAS ND)
3) Create a Deployment Manager profile
   - Using profile management tool
4) Install HTTP server
5) Setup Portal Node to communicate with the Deployment Manager
6) Run Portal federation Scripts
7) Run script to add the HTTP server to the cell
   - Copy the file "<plugin_root>/bin/configurewebserver1.bat" to "<dmgr_profile>/bin"
   - Run configurewebserver1.bat – user <was_admin_user> -password password
8) Manage multiple WAS instances using Deployment Manager
   - <dmgr_profile>/bin/startManager.bat or stopManager.bat

Step by step guide to setup Portal 6.1.5 cluster - http://www-01.ibm.com/support/docview.wss?uid=swg21413946

Federating Portal Node into a Cell – Overview of steps

1) Collect java classes for Deployment Manager (DM)
   - ConfigEngine.bat collect-files-for-dmgr -DwasPassword=password

2) Extract files from the generated zip file on DM
   - eg: WAS root @ c:\IBM\WebSphere\AppServer
   - This is because when you deploy new portlet to a portal server, the portal first uploads config changes and artifacts to DM, and then DM distributes files

3) Restart the Deployment Manager

4) Edit appropriate settings in wkplc.properties, wkplc_comp.properties
   - eg: IP, ports, user ids required by DM, as well as DB passwords

5) Run the pre-federation script
   - ConfigEngine.bat cluster-node-config-pre-federation -DdmgrUserid=uid=wasadmin,o=defaultWIMFileBasedRealm -DdmgrPassword=wasadmin

6) Verify settings in wkplc.properties, and run the post-federation/cluster setup task
   - ConfigEngine.bat cluster-node-config-post-federation
   - ConfigEngine.bat cluster-node-config-cluster-setup

Portlet Deployment in a Cluster

- 3 available methods to deploy portlets
  - Copy deployed portlet EAR from staging portal to production portal:
    - Use XMLAccess to export the portlet definition at staging server
    - Use XMLAccess to import the portlet definition at production server
  - Deploy as an EAR across the cluster:
    - Deploy using WAS console or wsadmin script
    - Use XMLAccess to import the portlet definition
  - User the Manage Web Module portlet
    - Log into any cluster member (the cluster node don't need to be stopped)
    - Install the portlet
- Synchronise all cluster nodes
  - Default auto synchronisation is once per min
- Activate the portlets, using either methods:
  - Activate individual portlets using the Manage Web Modules portlet
  - Activate all portlets using the ConfigEngine task "*activate-portlets*"

Two different approaches

1. Step 1 → Step 2 → Step 3

Deploy: PortalUI or XMLAccess from any node

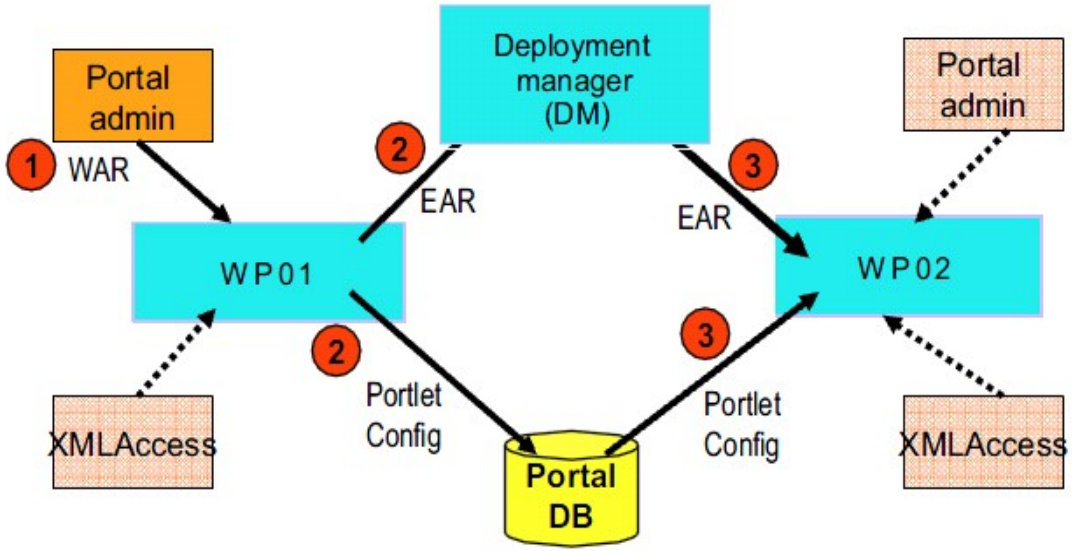
Synchronize: With all nodes

Activate: Using PortalUI or XMLAccess from any node

2. Step 1 → Step 2

Deploy: PortalUI or XMLAccess from any node

Run config task to activate: **activate-portlets**

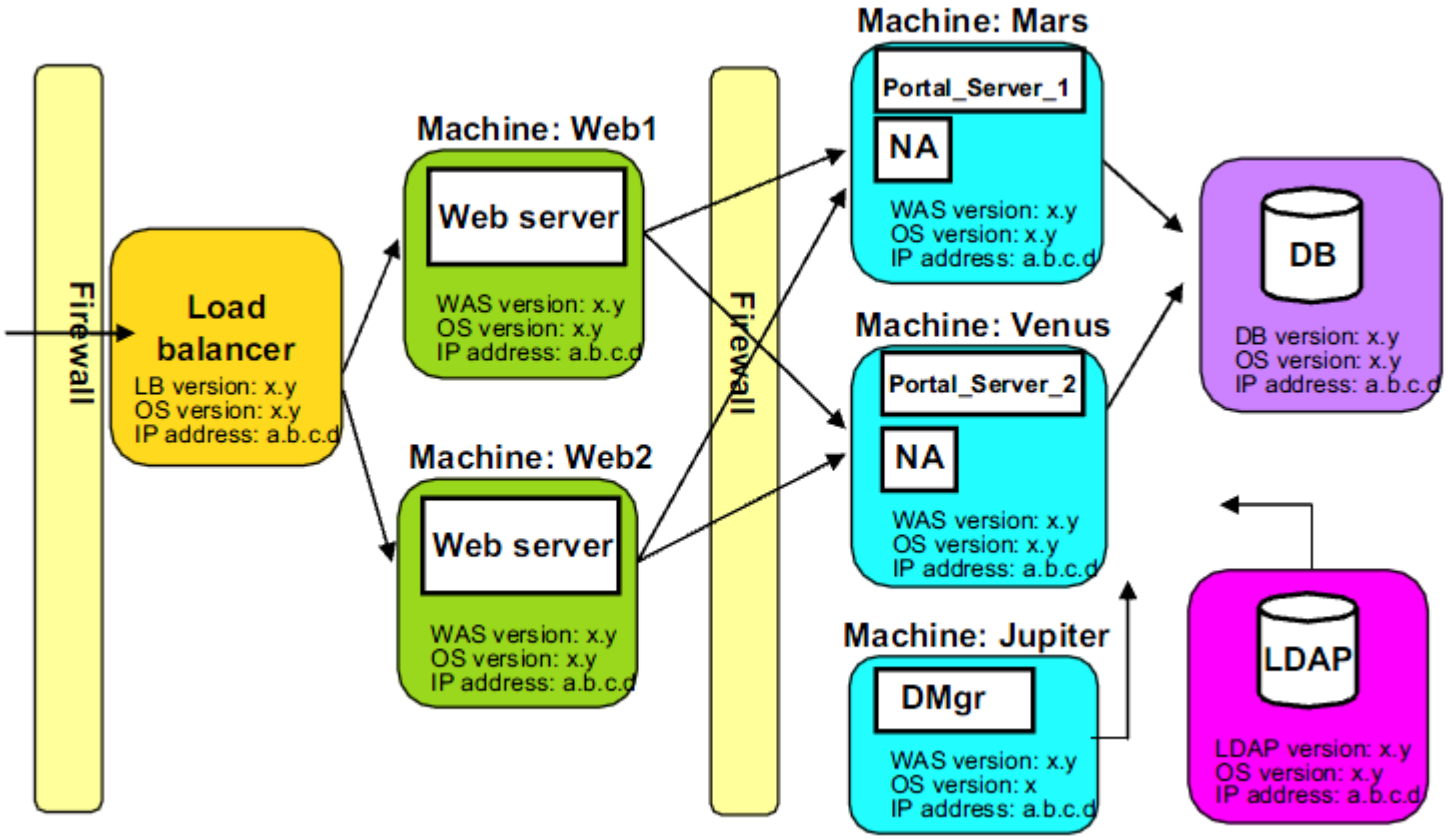


Troubleshoot of Portal Cluster

Problem Determination Procedure

1) Before a problem occurs
   - Problem prevention
   - Prepare for problem management – Topology diagram, diagnostic data collection checklist
2) Organise the investigation
   - Characterise the problem
   - Table of issues and symptoms
3) Initial investigation – Phase 1
   - Address key issues/symptoms, and research in knowledge base
   - Address an issue at a time
4) In depth investigation – Phase 2
   - Identify problem category
   - Find/execute specific troubleshooting actions
   - Leverage IBM Support Assistant (ISA, www.ibm.com/software/support/isa)

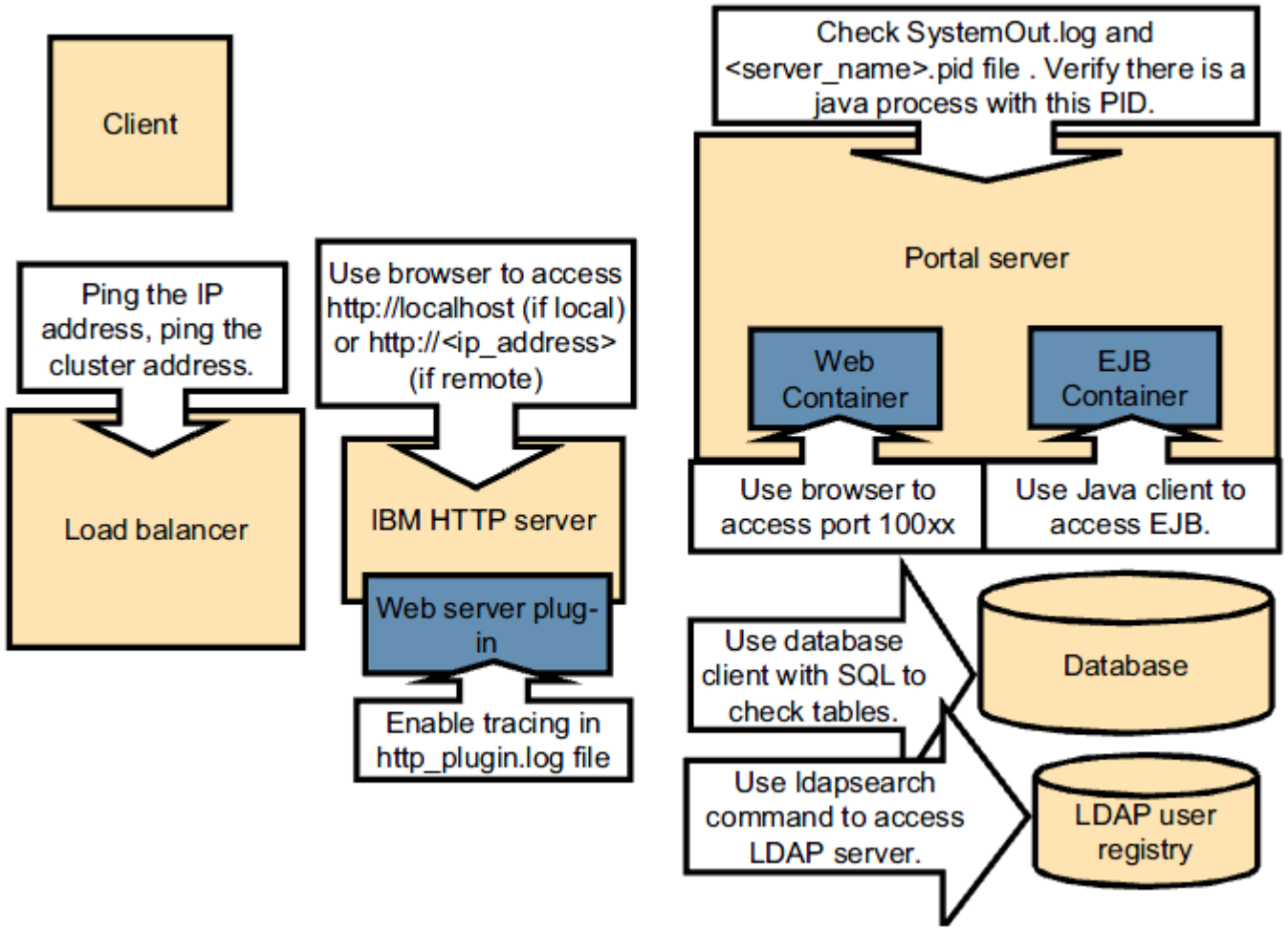
Sample Topology Diagram



Relevant Log Files

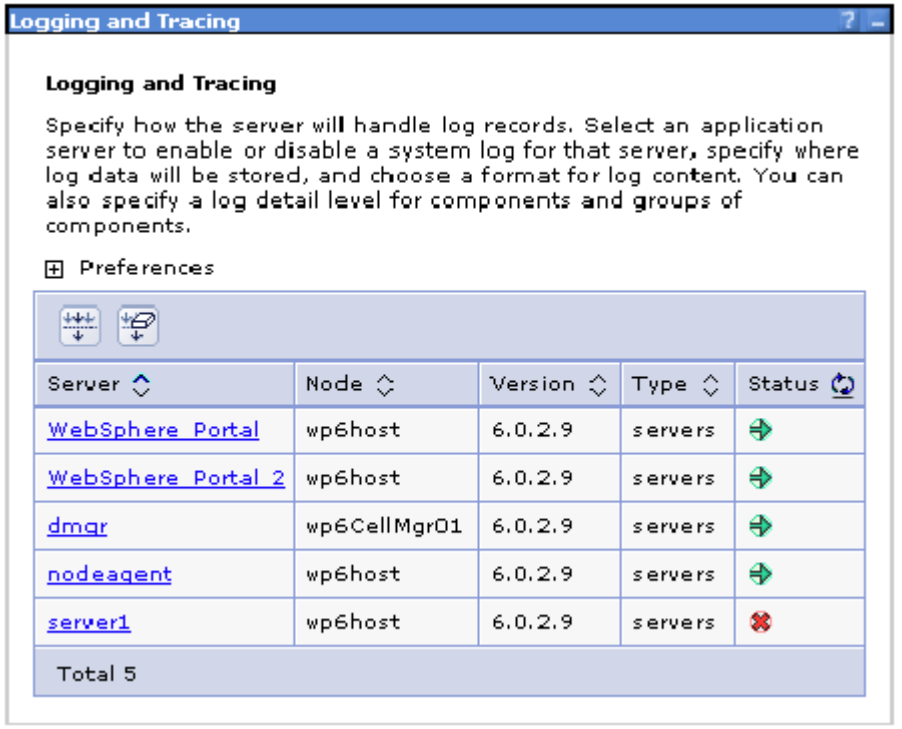
- Portal Installation logs
  - <wp_root>\log\
- Configuration logs
  - <profile_root>\ConfigEngine\log
- WebSphere logs
  - <profile_root>\logs\server1
  - <profile_root>\logs\WebSphere_Portal
- Deployment Manager Directory
  - <dmgr_profile_root>\logs\dmgr
- HTTP Plugin logs
  - Examine httpd.conf file for Plugin installation location

Isolate problematic component(s)



Enable Portal Tracing

- In WAS console (for all servers)
- Go through to Troubleshooting → Logs and Trace → server_name
- Click on Diagnostic Trace → Log Format
- Set log detail levels



And Trace Portal Problems...

- Areas of concern include:
  - Access control (com.ibm.wps.ac.*)
  - Authentication (com.ibm.wps.services.authentication.*=all)
  - Database (com.ibm.wps.datastore.*=all)
  - Personalisation (com.ibm.websphere.personalisation.*=all)
  - Portal Search (com.ibm.portal.search=all)
  - Portlet container (com.ibm.wps.pe.pc.*=all)
  - Services (com.ibm.wps.services.*=all)
  - XML configuration interface (com.ibm.wps.command.xml.*=all)

10 Portal Debugging Tips

1) For problems in initial installation and configuration process, check these files:
   - <wp_root>\log\wpsinstallog.txt
   - <wp_root>\log\ConfigTrace.log
2) Check the HTTP server plug-in log to verify that there are no problems with request routing
   - <plugins_root>\logs\<webserver_name>\http_plugin.log
   - Enable the Trace setting for the http_plugin.log
3) Isolate the problem to a specific cluster member
   - Use the cluster member's internal HTTP port
   - Does it happen on all cluster members, or just one?
4) Check the cluster member SystemOut.log and SystemErr.log
   - Includes both WebSphere Application Server and Portal messages
   - Default location of logs for secondary nodes is the <profile_root>/logs/<server_name>directory

10 Portal Debugging Tips

5) Use the deployment manager administrative console to:
   - Validate configuration
     - **Troubleshooting -> Configuration Validation -> Error**
   - Examine runtime messages
     - **Troubleshooting -> Runtime Messages**
6) For portlet deployment and synchronisation errors, also check node agent logs and deployment manager logs
   - Default location for node agent logs is the <profile_root>\logs\nodeagentdirectory
   - Default location for deployment manager logs is the <profile_root>\logs\dmgrdirectory
7) Enable tracing on any components that throw exceptions.
8) Use ISA Search component to research knowledge repositories
   - APARs, tech notes, information centres, and so forth
9) Use ISA collector tool to gather information on system and problem type.
10) Open a PMR with IBM Support
    - Submit PMR through ISA, along with output (JAR file) from the collector tool.a

Portal Security Overview

WebSphere Portal Security - Several Broad Subjects

- Who you are (authentication)
  - What is the user registry? What is the authentication mechanism?
  - WebSphere Portal includes a custom registry and supports multiple LDAPs and External Security Managers
  - "Lookaside" database for managing attributes outside LDAP
  - Single Sign-On
    - Portal participates in SSO with other applications
    - Portal includes a "Credential Vault"
    - Portlets can pass your identity credentials to a back-end application
- What you can do (authorization)
  - WebSphere Portal permissions (Portal Access Control)
  - External Security Managers, e.g., Tivoli® Access Manager, SiteMinder
- User Management (Quasi related to security)

Portal Authentication

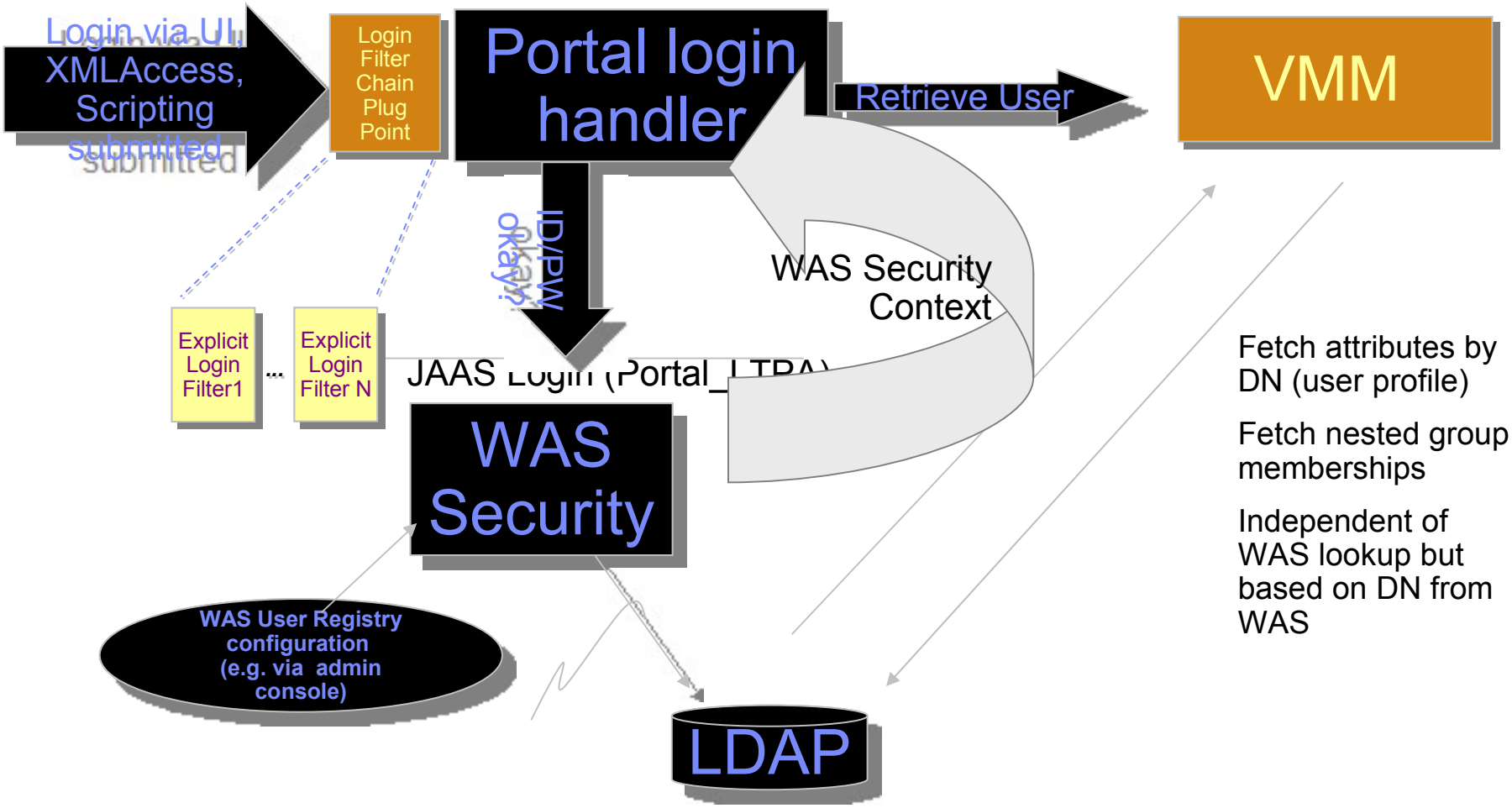
Portal Authentication

- WP is a custom Form Login application to WAS
  - relies on WAS to
    - intercept requests to protected portal area
    - do the authentication and provide the security context
    - Global Security in WAS is active

- Portal picks up whatever user identity established by WAS
  → All WAS authentication customization options also apply to portal:
  - Authentication Proxies and Trust Association Interceptors (e.g. TAM / WebSeal)
  - Custom JAAS Login Modules

- Portal supports public code plug points for intercepting the portal login and session validation flow

Portal and WAS Authentication "flow" (since version 6.1.x)

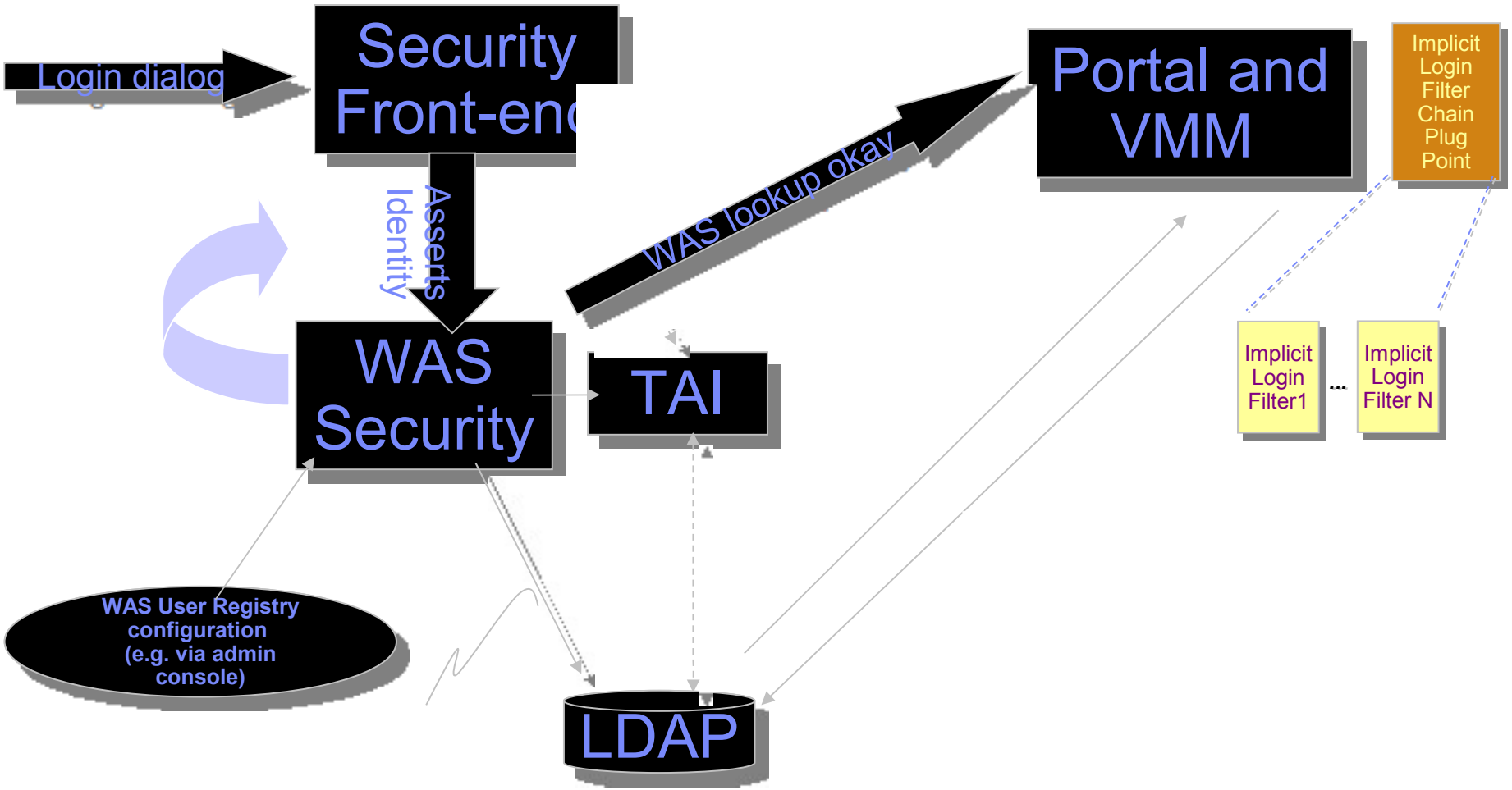


Fetch attributes by DN (user profile)

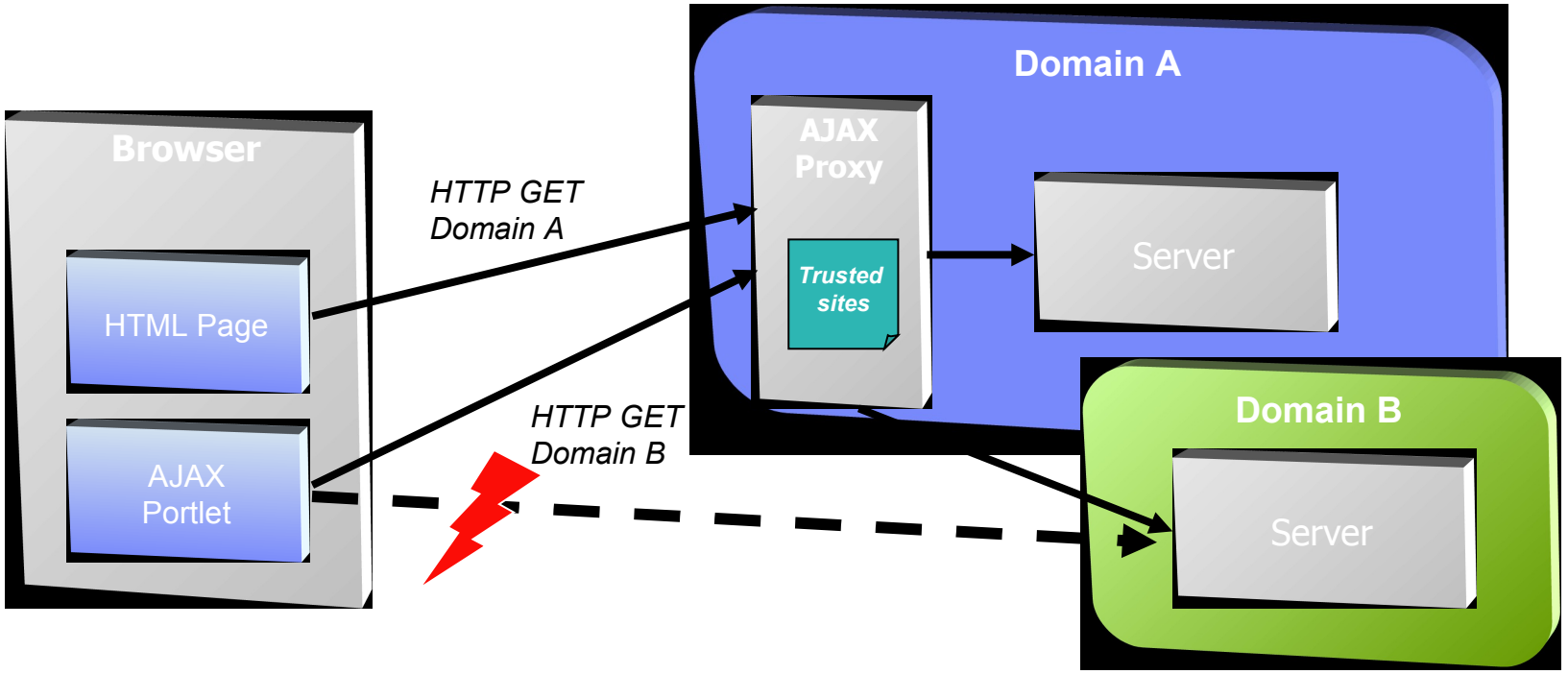
Fetch nested group memberships

Independent of WAS lookup but based on DN from WAS

Portal and WAS and TAI Authentication "flow"

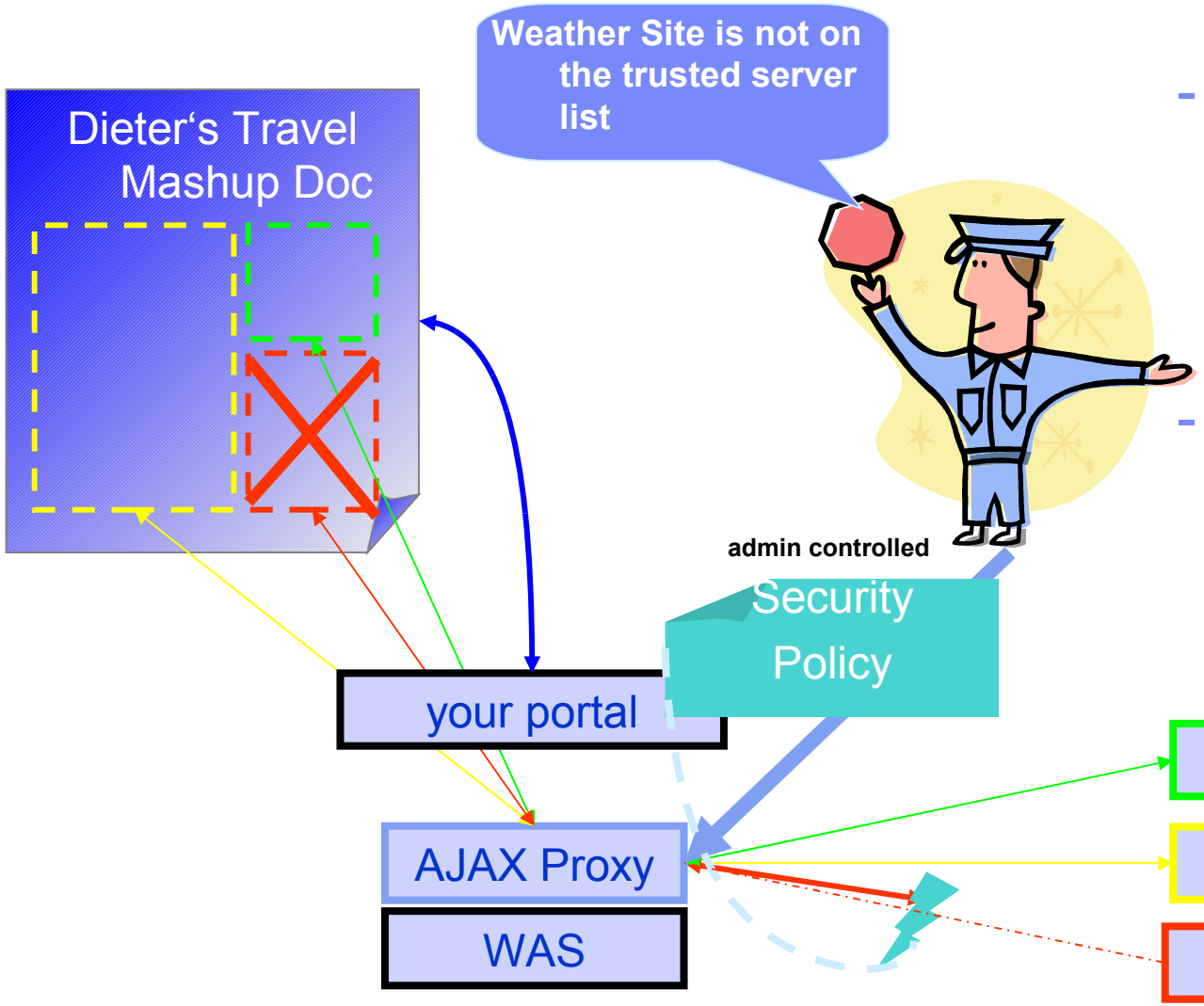


Portal AJAX Proxy



- Today's browsers prevent asynchronous requests to foreign domains because of security reasons.
  - Example: Your portlet is served from www.mycompany.com but your AJAX application tries to load a feed from cnn.com. This would be blocked by the browser

How Portal AJAX Proxy Server help guards it

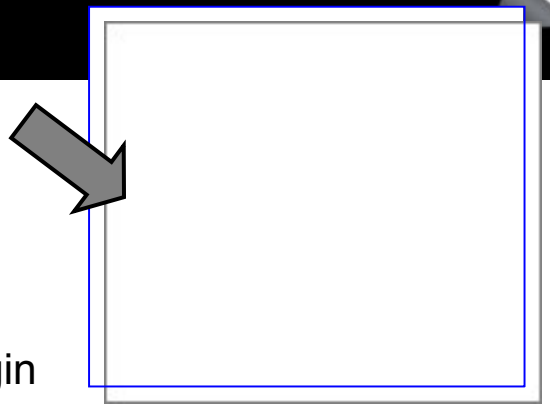


- Filtering based on
  - URL
  - HTTP Action
  - mime-type
  - Requesting user
- Maintained by Security Administrator

New in 6.1.: User Profile REST Service

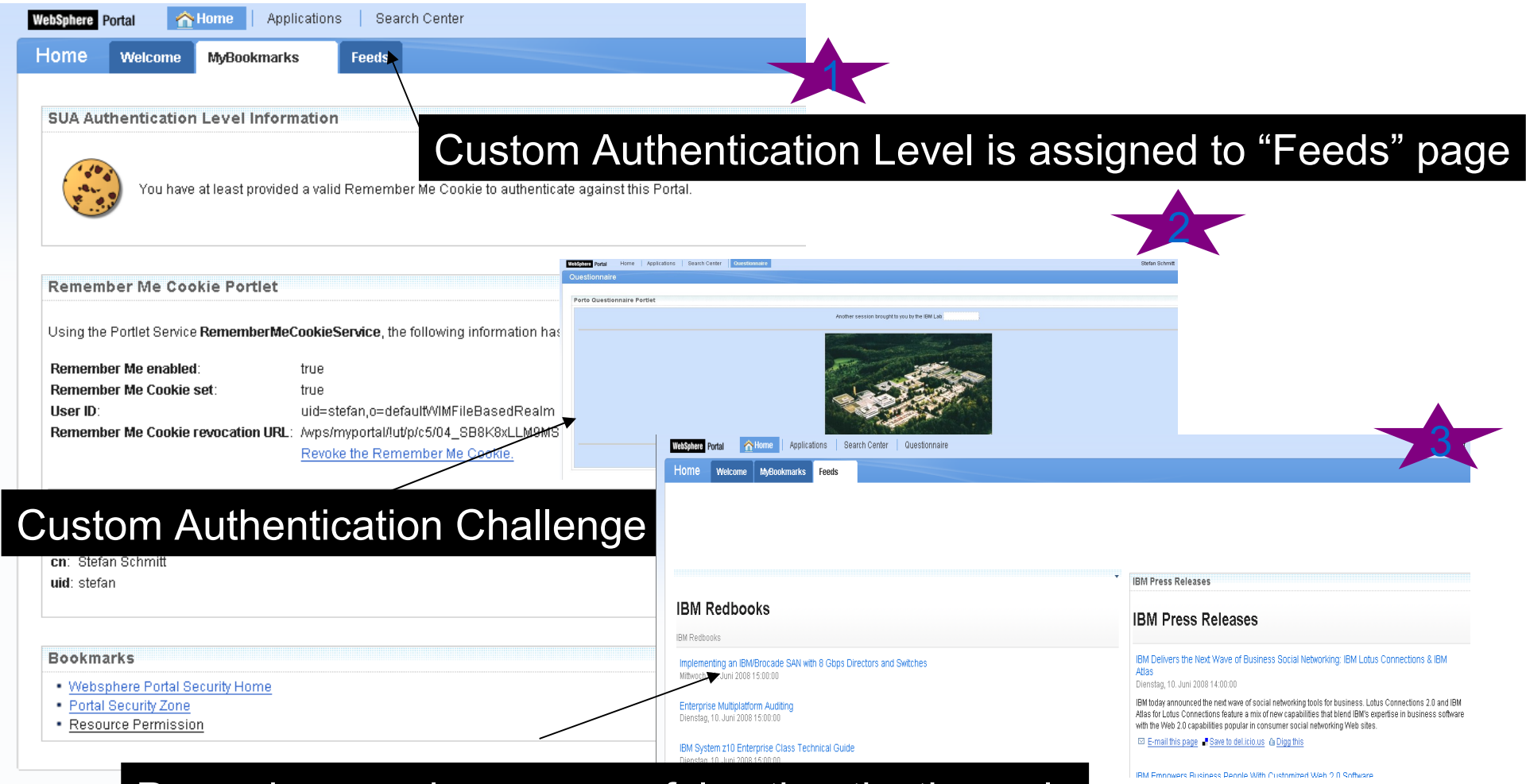
- Provides ATOM feeds for
  - Defined user/group attributes
  - User/group profiles
  - User/group searches
  - Group membership
- Supports CRUD operation through ATOM Publishing Protocol (APP)
  - Create user/group
  - Delete user/group
  - Update user/group profile
  - Add user to group
  - Remove user from group
- Supports virtual portal realms

StepUp and RememberMe

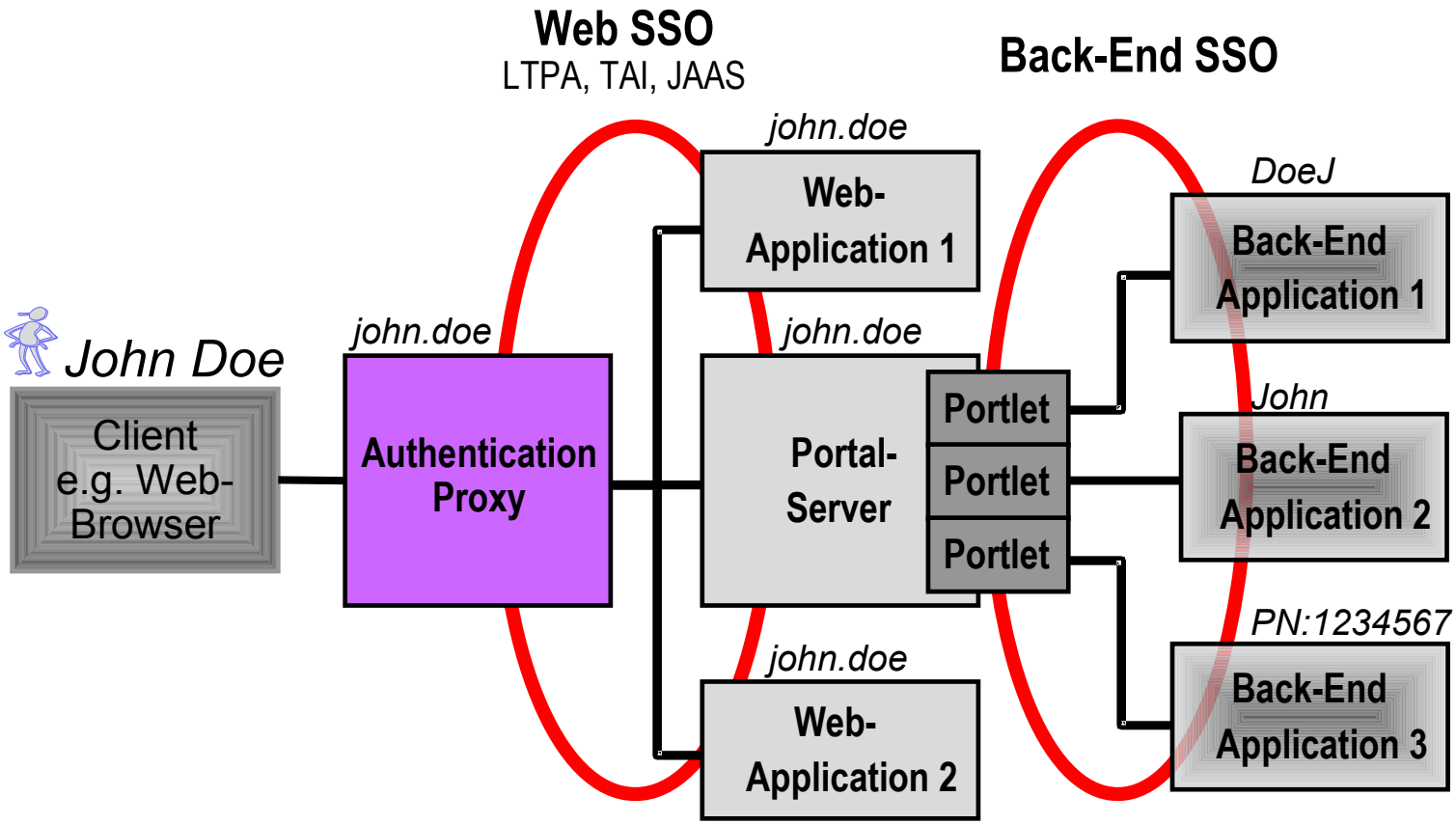


- **RememberMe Cookie**
  - Persistent cookie allows portal to recognize user without login
    - → Portal can show a personalized welcome page
  - If cookie is present, portal treats the user as **„identified“** but not yet **„authenticated“**
    - User can only see resources available for the anonymous user
  - Access to protected resources requires the user to authenticate.
- **StepUp Framework**
  - Enables you to enforce another authentication level for specific pages and/or portlets
  - Framework to **plug in additional Levels of Authentication Strength**
    - E.g. enforce SSL for specific servcies, or client cite certificates,…
  - Available for Pages and Portlets
  - Required authentication strength can be managed using the Resource Permission Portlet

Custom Authentication Level Sample – Step-Up

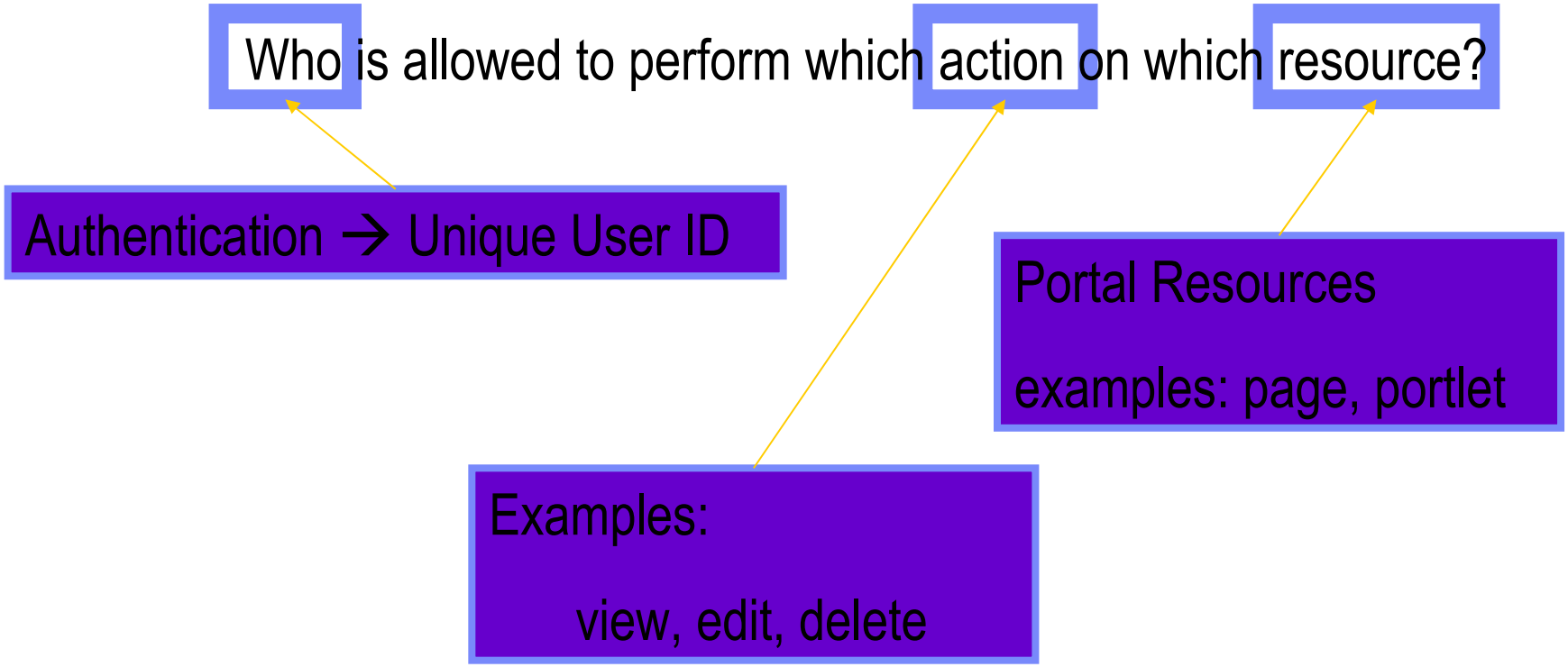


Portal Single Sign-On Realms



Portal Authorisation

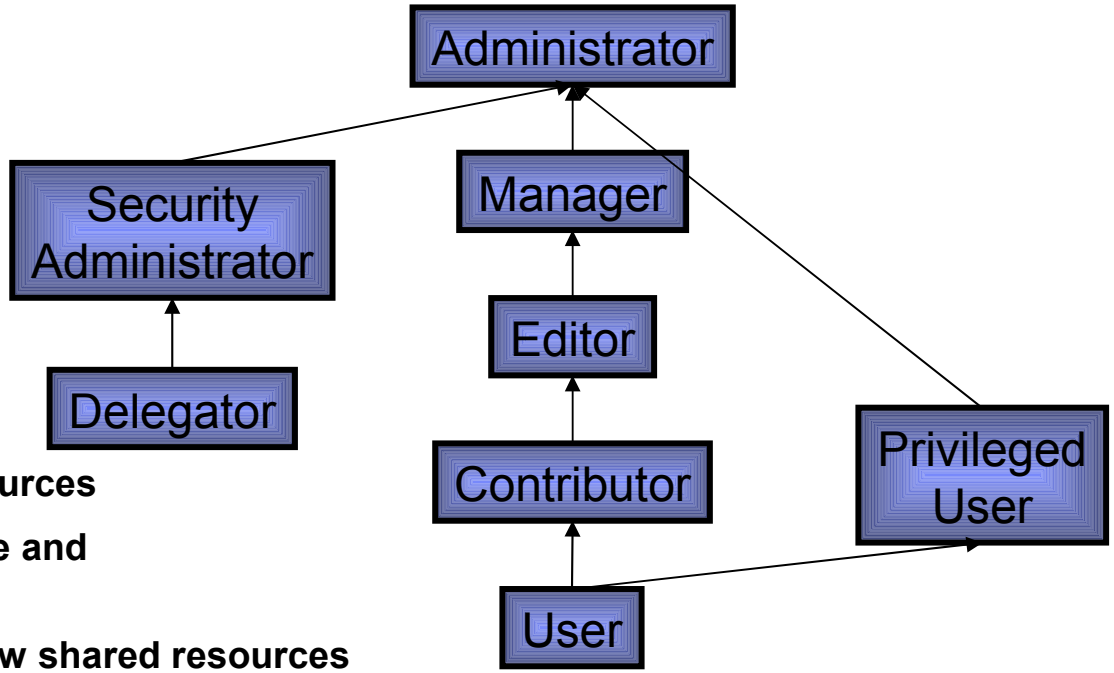
What is Access Control (aka. Authorization)?



Role Based Access Control

- Access control is based on roles
  - A role defines the permissions that a user or group has on a resource
  - Resources are pages, portlets, administration, Web Content Management (WCM) libraries, etc.
- Roles can be explicitly assigned to users or groups…
- Or, inheritance allows permissions contained in a parent resource to automatically propagate along a resource hierarchy
- Role Blocking
  - Inheritance Blocks stop a resource from inheriting parent permissions
  - Propagation Blocks stop inheritance from the parent resource to its children
- New access control and modifications can be done dynamically in real time
  - Users may need to log off and log in to see changes
  - Portal caches user permissions at login to improve performance

Portal Role Types



- **Users are allowed to view portal resources**
- **Privileged Users are allowed to create and personalize private resources**
- **Contributors are allowed to create new shared resources**
- **Editors are allowed to create and edit shared resources**
- **Managers are allowed to create, edit, and delete shared resources**
- **Delegators are allowed to grant access to other principals**
- **Security Administrators are allowed to grant access on a resource to other principals**
- **Administrators are allowed to do everything**

Access Control Administration

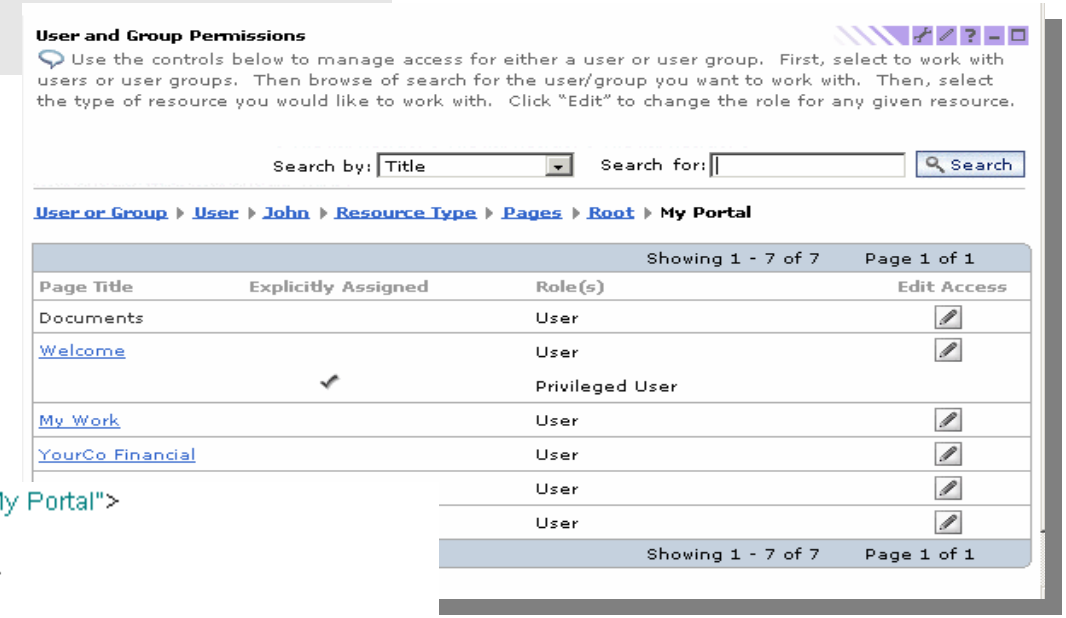
```
#Get the PacList of the Content Node you want to change
$Content find page uniquename ibm.portal.SamplePage.2 select
$PacList edit [$Access getacl Content [$Content current]]

#Now modify the AccessControl setting
$PacList grant User name "uid=youruser,dc=users,o=myorg"
$PacList grant Manager name "uid=wpsadmin,dc=users,o=myorg"

$PacList block User inheritance
$PacList block Manager inheritance
```

Portal Scripting

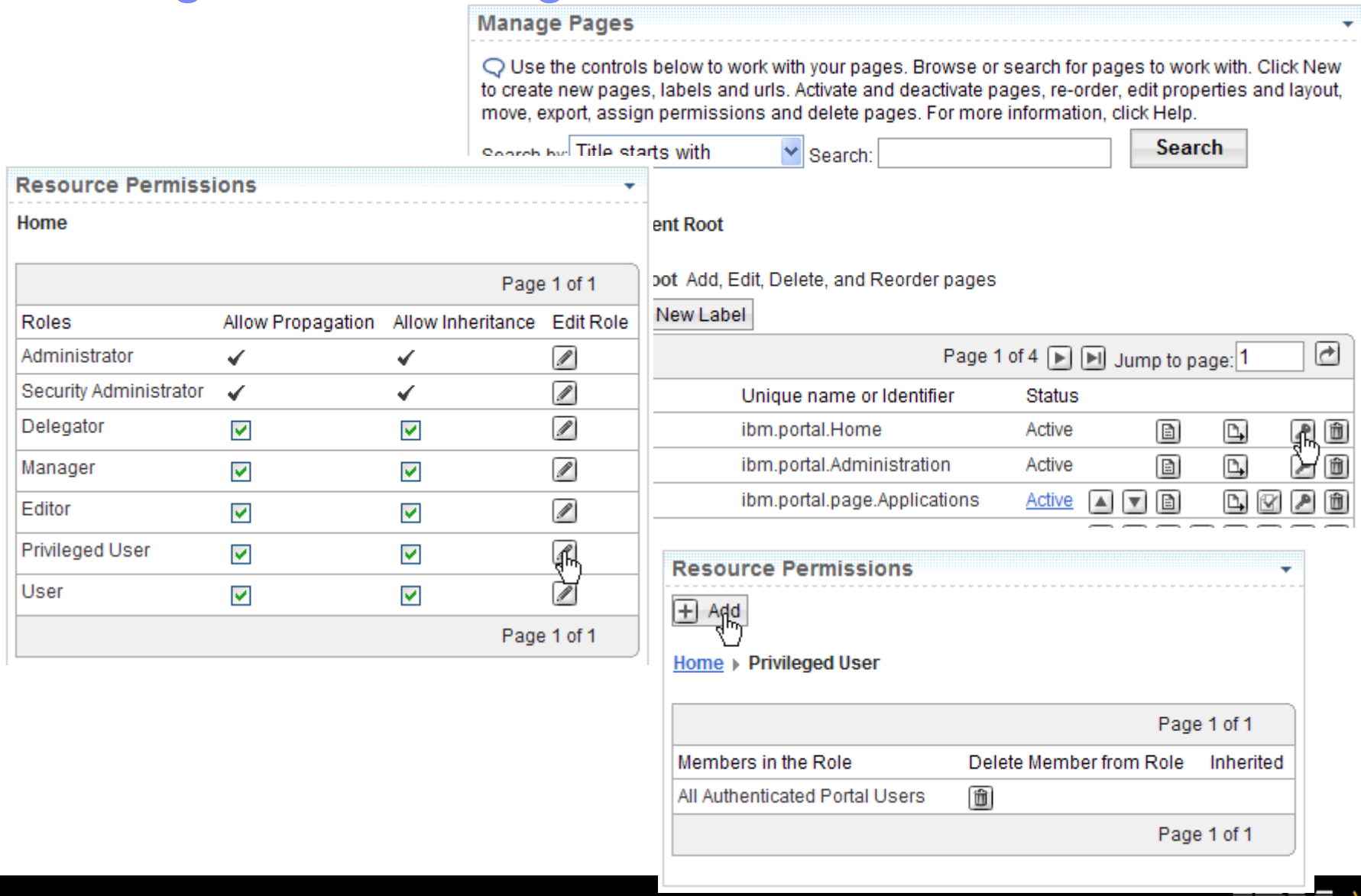
Administration Portlets




```
<content-node action="update" uniquename="wps.My Portal">
 <access-control>
  <role-block type="propagation" actionset="User"/>
   <role actionset="Administrator" update="set">
     <mapping subjectid="uid=hsimpson,o=default" subjecttype="USER" update="set"/>
   </role>
   <role actionset="User" update="set">
     <mapping subjectid="anonymous portal user" subjecttype="USER" update="set"/>
   </role>
  </access-control>
</content-node>
```

XmlAccess

Granting Access Using Administration Portlets



WebSphere Portal Security Strategy

- Security is part of the Portal Design Process
  - Design documents detail on security implications and are reviewed by dedicated portal security team
- Security is part of the Portal Testing Strategy
  - WebSphere Portal does dedicated security vulnerability (aka. penetration) testing on selected portal releases
- Security Certifications
  - WebSphere Portal access control is Common Criteria certified
  - WebSphere Portal uses FIPS 140-2 compliant crypto libraries provided by WebSphere Application Server
- Portal Security is aligned with IBM Security Strategy
  - IBM invests in security research, e.g. teams in Zurich and Tokyo investigate Web20 security implications
  - WebSphere portal security architect aligns portal security with IBM security strategy and research results
- Security fixes are published on the portal Security Bulletin Web Site

  http://www-128.ibm.com/developerworks/websphere/zones/portal/security/

धन्यवाद
Hindi

多謝
Traditional Chinese

ขอบคุณ
Thai

Спасибо
Russian

Gracias
Spanish

Thank You
English

Dziękuję
Polish

شكرا
Arabic

Obrigado
Brazilian Portuguese

多谢
Simplified Chinese

Danke
German

Grazie
Italian

Merci
French

நன்றி
Tamil

ありがとうございました
Japanese

감사합니다
Korean