How to…

Implement High Availability for SAP Enterprise Portal 6.0

SAP ENTERPRISE PORTAL 6.0

PUBLIC

ASAP "How to…" Paper

Applicable Releases: EP 6.0 SP2 (Patch 3) with J2EE Engine PL21

April 2004

Document version: 1.0

Copyright

Icons in Body Text

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see Help on Help → General Information Classes and Information Classes for Business Information Warehouse on the first page of any version of SAP Library.

Typographic Conventions

Type Style	Description
Sample text	Words or characters quoted from the screen. These include field names, screen titles, pushbutton labels, menu names, menu paths, and menu options. Cross-references to other documentation.
`Sample text`	Emphasized words or phrases in body text, graphic titles, and table titles.
Sample TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Sample text`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Sample text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Sample text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`Sample TEXT`	Keys on the keyboard, for example, F2 or ENTER.

CONTENTS

1 Goals of High Availability Planning 4
2 Factors to Consider 4
2.1 Planned and Unplanned Downtime 4
2.2 Determining Single Points of Failure 5
2.3 Types of Clustering 6
2.4 Security 7
2.5 Decision Making 7
3 System Components in High Availability Planning 8
3.1 J2EE Engine 9
3.2 Portal Platform 11
3.3 Persistence Layer 13
3.3.1 Portal System Database 14
3.3.2 User Persistence Store 15
3.3.3 Central Configuration Data Directory 15
3.4 Search and Classification (TREX) 15
3.5 Unification 16
4 High Availability Scenarios 17
4.1 Highly Available SAP Enterprise Portal without Database Clustering 17
4.2 Highly Available SAP Enterprise Portal with Database Clustering 18
4.3 Highly Available SAP Enterprise Portal with Database and TREX Clustering 19
4.4 Highly Available and Secure SAP Enterprise Portal 20
5 Case Studies 21
5.1 HP-UX Internal Case Study 21
5.2 International Company A 23
5.3 International Company B 24

1 Goals of High Availability Planning

SAP Enterprise Portal 6.0 unifies applications, information and services of an enterprise on a single system. To ensure that the portal runs with minimal downtime, the main systems in the portal landscape can be optimally set up so that none of them is a potential point of failure.

This guide outlines the factors to consider when planning a highly-available portal landscape for **SAP Enterprise Portal 6.0 SP2**. It offers solutions at each level of portal deployment and provides sample scenarios.

Before you begin, you should review the following documents; they provide the needed background in SAP Enterprise Portal 6.0 architecture and general SAP high availability guidelines:

- *SAP Enterprise Portal Technical Infrastructure,* which is part of the installation documentation for SAP Enterprise Portal 6.0 SP2. It is accessed from SAP Service Marketplace at **`service.sap.com`** → *Documentation & More* → *Installation* → *EP6.0 SP2 Install Guide*.
- *High Availability for mySAP.com Solutions,* a white paper accessed from SAP Service Marketplace at **`service.sap.com/ha`**. This URL should also serve as a starting point for accessing high availability information in general.

All information and references about SAP J2EE Engine 6.20 relate to the SAP J2EE Engine 6.20 Patch Level 21.

2 Factors to Consider

2.1 Planned and Unplanned Downtime

Two types of **downtime** need to be taken into consideration for high availability planning:

- **Planned downtime**
  With planned downtime, the system is not available for production due to planned maintenance tasks such as network maintenance, system and application maintenance, system upgrades and database backups. For customers requiring 24 x 7 availability, these downtimes must be either eliminated or kept as short as possible.

  Planned downtime in the portal landscape can be reduced either by implementing measures that significantly decrease downtime (during upgrades and transports, for example) or by performing procedures online (such as backup and changing system profiles). For more information, see SAP Service Marketplace at **`service.sap.com/ha`** → *High Availability* → *HA in Detail* → *Planned Downtimes*
- **Unplanned downtime**
  Failures occur due to hardware, system software and infrastructure problems, as well as operator errors. The result is that system operation is interrupted until the point of failure is located and the error is corrected.

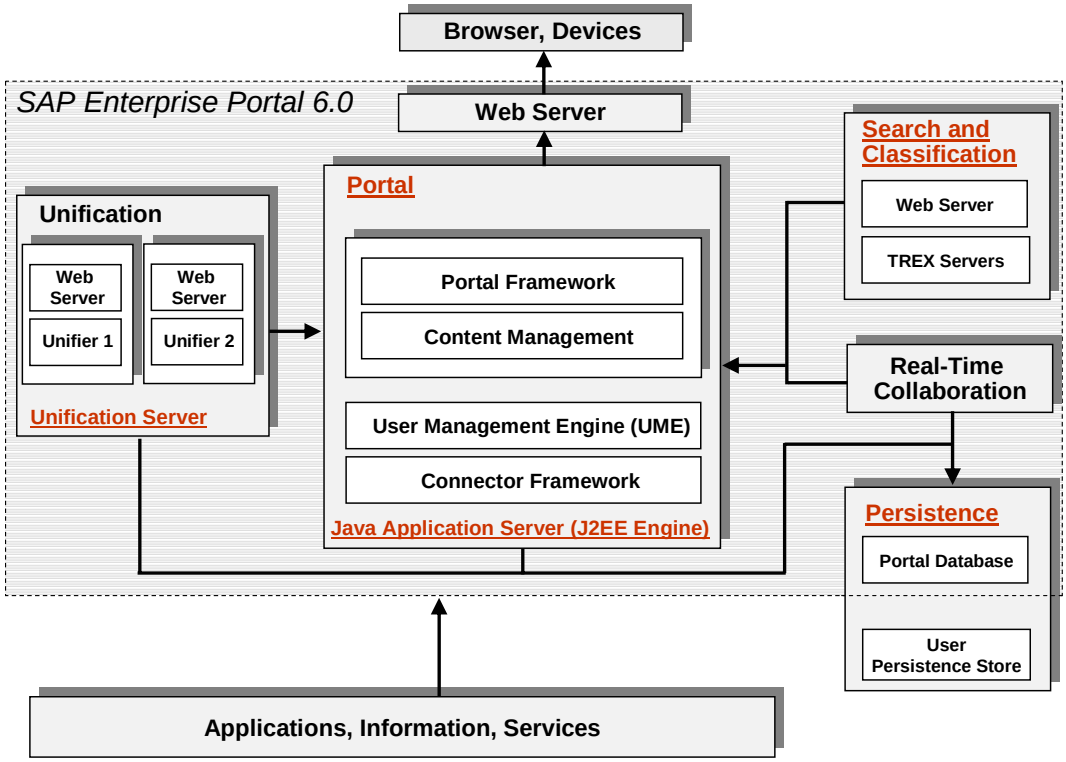
  Unplanned downtime can be minimized by determining each potential single point of failure in the entire system, understanding the interaction between the components, and implementing recommended high-availability measures at each level accordingly. This document describes the components of SAP Enterprise Portal that can be configured for high availability.

Disaster planning is one type of unplanned downtime that results from a disabling event, such as a fire. In this case, a disaster recovery plan must be formulated to ensure that the installation recovers its computing services as quickly as possible. The disaster recovery strategy must include a backup and restore plan that determines which data should be backed up and the procedures that will be used to restore it. For more information about disaster recovery, see *High Availability for mySAP.com Solutions*. For information about backup and restore, see the *Solution Management Guide* at SAP Service Marketplace at **`service.sap.com/ep60`** → *Documentation & More* → *Fundamentals* → *EP 6.0 SP2: Solution Management.*

2.2 Determining Single Points of Failure

Portal availability increases as more points of failure throughout the portal landscape are eliminated. These single points of failure can be removed by using redundant components to avoid unplanned downtime.

The technical components of SAP Enterprise Portal 6.0 are described in the following figure:



The main potential points of failure in the portal (underlined in the diagram) are:

- J2EE Engine
- The portal itself, operating onJ2EE Engine
- Persistence layer, containing the portal database and the user persistence store
- Search and Classification (TREX)
- Unification Server

For each single point of failure, a solution must be implemented to guarantee continuous operation of the portal. Various types of cluster and switchover solutions are available for the different components: some are standard proven solutions provided by SAP hardware partners and others are manufacturer-specific and proprietary. Most of these solutions are based on clustering and load balancing.

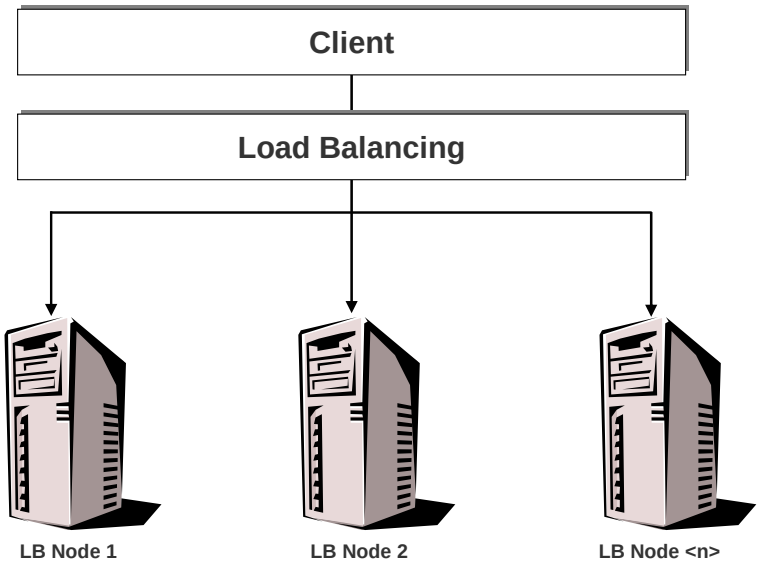
2.3 Types of Clustering

With clustering, two or more servers operate together as a single logical system. Programs accessing services of the cluster need not know which cluster node is actually performing the work.

There are two general types of clusters:

- **Active/active clustering**
  External hardware or software load balancing can be used to distribute the workload between the connected servers and thereby achieve highly available, reliable performance of the cluster. With load balancing, client requests are distributed to the different nodes of the cluster according to the distribution policy – such as round robin or fewest connections – defined in the load balancer.

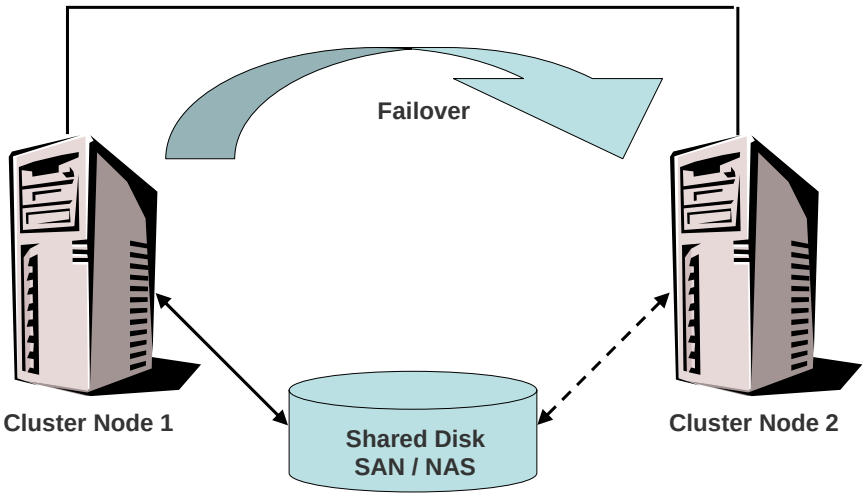
  For SAP Enterprise Portal, the load-balancing solution should ensure session persistence, or “stickiness.” This means that all requests of a specific user in a single session are forwarded to the same server. Stickiness is achieved at initial logon, by linking to the IP address and local port of the load-balancing system. If one node fails, the other servers assume the tasks of the failed node without any pause in operation or loss of information. The following figure illustrates an active/active cluster with load balancing:



- **Active/passive clustering**
  In this type of cluster, one node is the primary, active node, while the second — identical – node remains idle, in dedicated standby mode. If one node fails, the other machine takes over the task as soon as possible.

  Active/passive clustering is generally used with the database, in order to ensure reliable failover if errors are detected in the database server or if the server fails. If an error is detected, failover is performed. Once running, the standby node operates with minimal performance degradation evident to the client. To ensure seamless failover, an SAN (storage array network) or NAS (network attached storage) shared disk is used in the persistence layer.

The following figure illustrates an active/passive cluster.



2.4 Security

When planning a high-availability solution for a secure network, additional clustering and replication within and outside the demilitarized zone (DMZ) may be required. For further information, refer to:

- *SAP Enterprise Portal Technical Infrastructure Guide*
- The "Secure Communications" section in the *SAP Enterprise Portal Security Guide*, accessed from SAP Service Marketplace at **`service.sap.com/ep60`** → *Documentation & More* → *Fundamentals* → *EP 6.0 SP2: Security Guide*

2.5 Decision Making

SAP Enterprise Portal can be made highly available at every level of the system landscape. The exact configuration to be used is based on the need to balance the requirements for high availability with specific customer considerations such as security needs, plans for future scaling, and the cost of high availability equipment versus the cost of potential downtime.

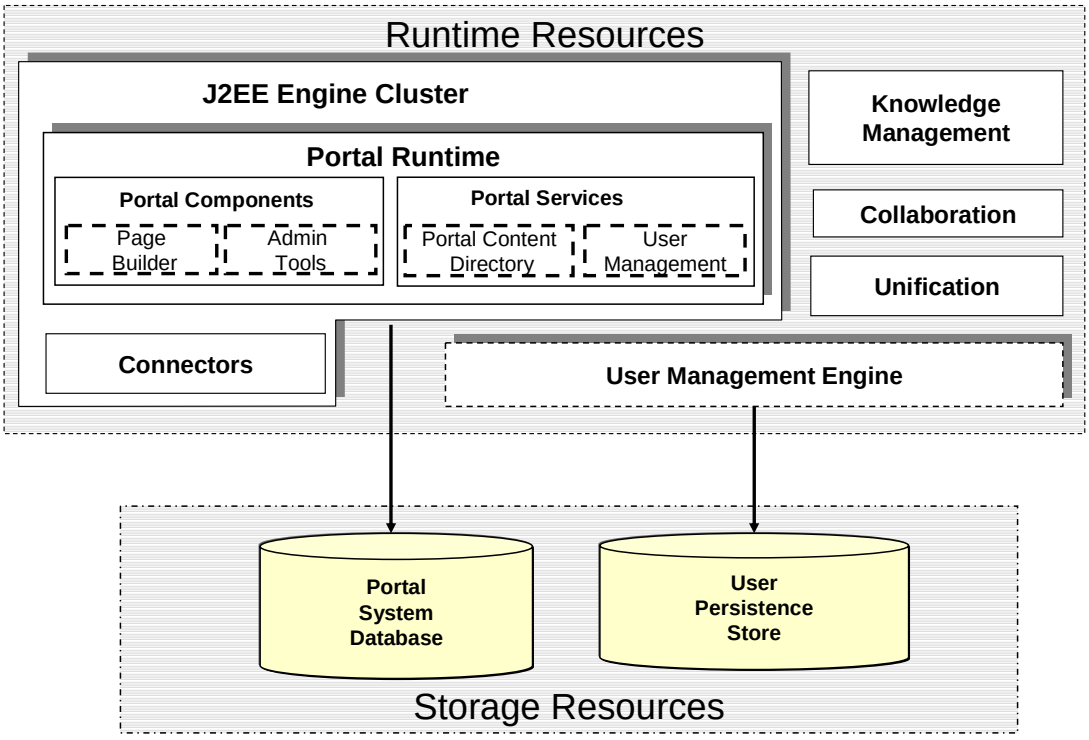
Documents that may assist in the decision-making process are:

- *SAP Enterprise Portal Technical Infrastructure Guide*
- *How to Perform Initial Sizing of SAP Enterprise Portal 6.0,* accessed from SAP Service Marketplace at **`service.sap.com/ep60`** → *Documentation & More* → *How-To Guides*
- *mySAP Enterprise Portal Solution Management Guide,* accessed from SAP Service Marketplace at **`service.sap.com/ep60`** → *Documentation & More* → *Fundamentals* → *EP 6.0 SP2: Solution Management*

3 System Components in High Availability Planning

This section describes the SAP Enterprise Portal components for which high availability planning should be considered. Refer to the *SAP Enterprise Portal Technical Infrastructure Guide* for details about the different configurations of the portal as a whole and the exact configuration of the specific components.

The basic runtime architecture of SAP Enterprise Portal can be depicted as follows:



For a full description of each component, see *Overview of SAP Enterprise Portal,* accessed from SAP Help Portal at **`help.sap.com/ep`** → *Administration, Developer and End User Documentation* → *SAP Enterprise Portal Documentation* → *Administration Guide* → *Cross Platform* → *Architecture of SAP Enterprise Portal.*

The following systems in the portal landscape – described previously as potential points of failure – should be considered for high availability planning:

- J2EE Engine
- Portal, running on J2EE Engine
- Persistence layer: the portal database and the user persistence store
- Search and Classification server (TREX)
- Unification Server

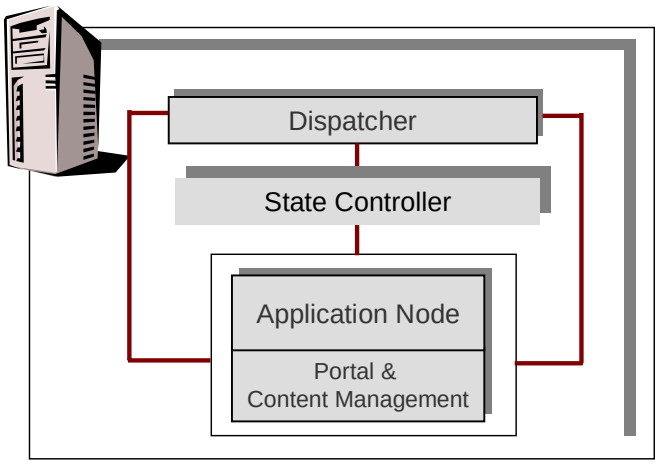
3.1 J2EE Engine

J2EE Engine is an application server that implements the J2EE standard, supporting servlets, Java Server Pages (JSPs), Enterprise Java Beans (EJBs) and other services. J2EE Engine PL21 features internal load balancing, high availability, reliability, and scalability through its three types of cluster elements: dispatchers, application nodes and a state controller:

- One or more **dispatcher nodes** dispatch requests to the application nodes and perform load balancing for even distribution of the load between participating application nodes.
- One or more **application nodes** handle the actual processing of user requests, received from the dispatchers.
- The **state controller** centrally manages the overall functioning of the entire cluster of application nodes. It serves as a single, persistent data storage area for maintaining ongoing configuration information. The state controller does not handle client requests but rather serves as the central, single source for synchronization between cluster elements.

  To ensure high availability, a second, backup state controller is recommended.

The most basic installation of a J2EE Engine consists of a single state controller, a single dispatcher and a single application node, as follows:



For more information about the SAP J2EE Engine cluster, refer to *Chapter 2: Administration of SAP J2EE Engine 6.20 Cluster* in *The Revised SAP J2EE Engine 6.20 Cluster Architecture* guide, and the *Administration Manual* installed with SAP J2EE Engine, in the following path: `<SAPj2eeEngine_install_dir>/docs/index.html`.

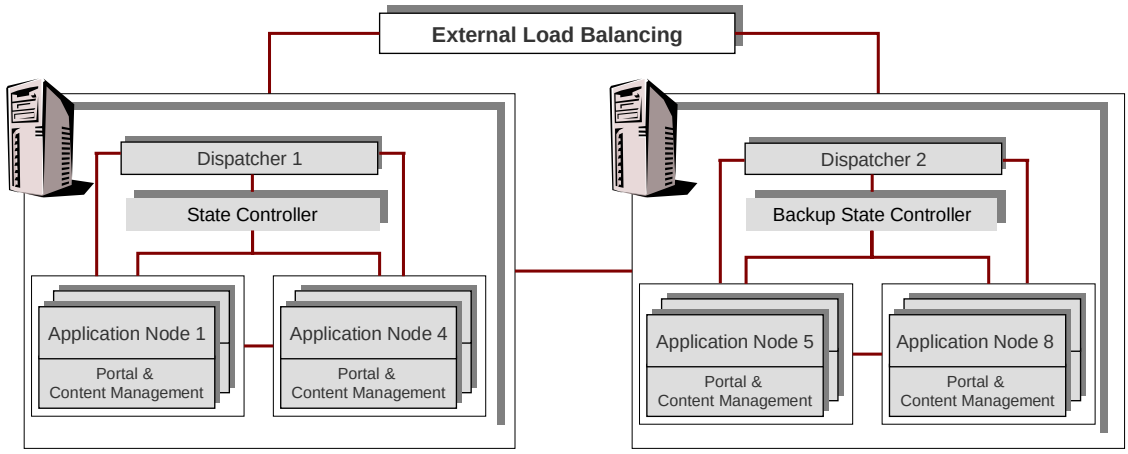
In order to prevent the state controller or dispatcher from becoming a single point of failure, high availability measures must be implemented at a number of levels:

- **To ensure persistent cluster data availability**, J2EE Engine provides an active/passive switchover solution, implemented by using a backup state controller installed in the cluster on a second machine. The backup state controller operates in standby mode, assuming the role of the state controller in case of failure.

- **To ensure process availability in the cluster,** high availability is implemented through element redundancy and load balancing. Scaling the cluster through multiple application nodes on each machine protects the cluster against software crashes: If an application node fails, the dispatcher on that machine – through use of an external load balancing mechanism – distributes client requests to other available application nodes. To guarantee high availability of the dispatcher function, several J2EE Engine instances are installed on multiple machines to ensure that in case of failure of one dispatcher, dispatchers on other machines will maintain ongoing operation of the overall cluster.

We recommend that you run the **Startup Framework** with your J2EE Engine / SAP Enterprise Portal cluster in order to optimize high availability of processes. This software controls the state of the different cluster elements and automatically restarts those elements that have failed for some reason. The Startup Framework helps maintain the planned level of process redundancy in the cluster in order to provide continuous availability of operations.

The following figure illustrates a J2EE Engine cluster on multiple hosts, with a third-party external load-balancing solution. Note that the first machine includes the state controller, while the second machine includes the backup state controller.



For more information about setting up and managing J2EE Engine, see:

- Switchover solutions in J2EE Engine: *SAP Web AS in Switchover Environments*, accessed from SAP Service Marketplace at **`service.sap.com/ha`** → *High Availability* → *Media Library* → *Documentation* → *Switchover*
- Configuring the J2EE Engine cluster for the portal platform: *How to Fine-Tune Enterprise Portal 6.0*, accessed from SAP Service Marketplace at **`service.sap.com/ep60`** → *SAP Enterprise Portal 6.0* → *Documentation & More* → *How-To Guides*
- Load balancing for J2EE applications: SAP Help Portal at **`help.sap.com`** → *SAP NetWeaver* → *SAP Web Application Server* → *SAP Web Application Server 6.20.* Then: *SAP Library* → *MySAP Technology Components* → *SAP Web Application Server* → *SAP J2EE Engine* → *Getting Started* → *Technical Overview* → *SAP J2EE Engine 6.20*
- Planning the cluster configuration: *SAP Enterprise Portal Technical Infrastructure Guide*

Rules to Follow:

For a J2EE Engine 6.20 PL21 cluster:

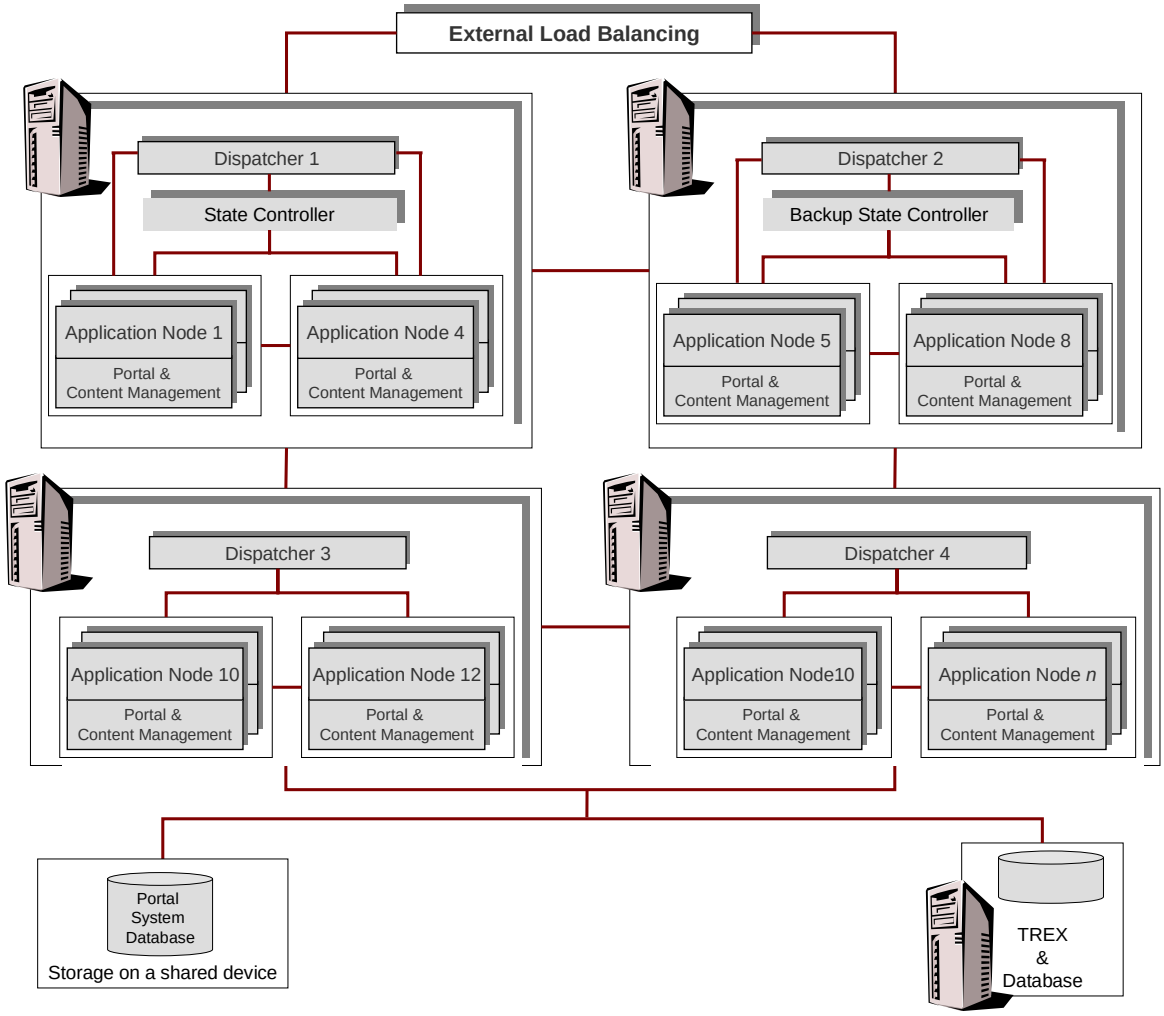
- Always designate at least one node as the dispatcher; otherwise, the cluster cannot operate. The order in which the application nodes and the dispatcher node are connected is not important.
- For high availability, **be sure to install a backup state controller on a separate machine** if possible. It is installed in addition to the state controller, so that if one fails, the other will assume its role.
- All machines on which the dispatcher and application nodes are started must be on the same Local Area Network (LAN). For SAP Enterprise Portal 6.0, we recommend using static host IP addresses.
- As a best practice, it is highly recommended to install all portal cluster nodes with the same J2EE engine instance number.
- Be sure that all identical servers of the J2EE engine nodes are of the same version.
- Generally, be sure to configure all identical services on the cluster nodes to have the same relation to external resources.
- All cluster nodes must be connected to the same J2EE cluster and reside in the same IP subnet. Separate J2EE clusters *cannot* access the same database repository.

3.2 Portal Platform

Based on the use of J2EE Engine clusters in conjunction with external load balancing, SAP Enterprise Portal can be clustered for high availability. To create the portal cluster using additional machines, you need to add an instance of the J2EE Engine cluster on each host. Follow the rules for implementing a state controller and backup state controller on separate machines, given in section 3.1, J2EE Engine.

When a hardware load balancer is used in a portal cluster configuration, the external access point to SAP Enterprise Portal (the URL IP address) is moved up from the J2EE Engine layer to the load balancer. This in essence moves the single point of failure up from server level to the load balancer, which should be made as highly available as possible, usually by replication. Consult with the vendor of the external load balancer to determine the most appropriate solution, which should implement a “sticky” load-balancing method based on either source-IP or cookies.

A large-scale portal cluster could be deployed as in the following figure:



Rules to Follow:

For the portal cluster:

- To guarantee proper portal functionality, external load-balancing solutions must always implement stickiness, using either a source-IP or cookie-based distribution policy.
- All machines in a portal cluster are covered by a license installed on any of them. However, for high availability purposes, you should install a license on at least two portal machines, although a license on each machine is recommended. In this case, if the machine with the single license is stopped for more than an hour (the period of time during which the license is cached), portal operation will not be interrupted.

To prepare the portal cluster for development and testing, and ultimately production, refer to the *SAP Enterprise Portal Technical Infrastructure Guide*.

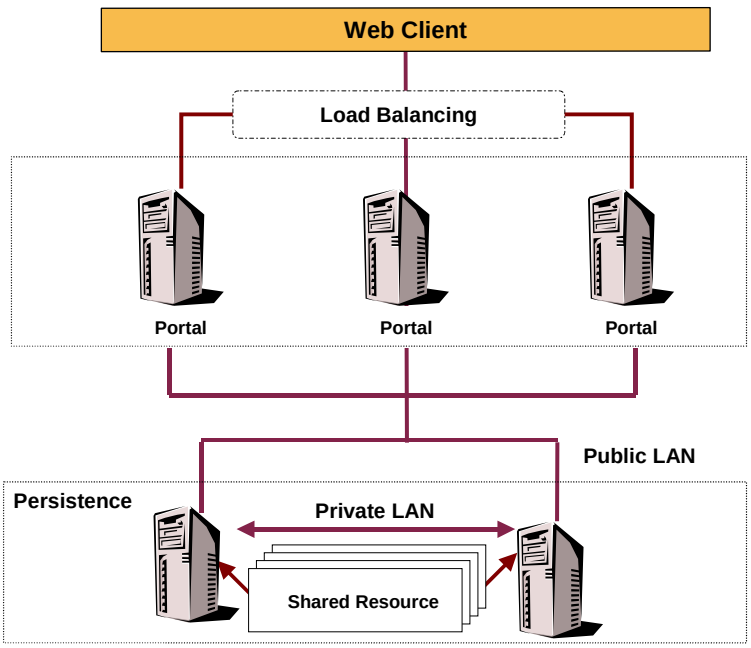
3.3 Persistence Layer

The persistence layer of the portal platform is the data repository needed by the portal during runtime. The persistence layer contains:

- The portal database, which consists of the Portal Content Directory (PCD), User Management Engine (UME) and Content Management schemas. It is used to store runtime objects such as role definitions and page-to-role relationships.
- User persistence store, which is the user data repository. It can be a database, a Lightweight Directory Access Protocol (LDAP) directory server or a SAP R/3 system.

To ensure high reliability of the persistence layer, it is possible to cluster the portal system database and user persistence store on a separate machine in an active-passive cluster, as in the following general figure:

Clustering the Persistence Layer



3.3.1 Portal System Database

Because the database is so crucial to portal operation, not only should it be safeguarded against failure, but the data itself must be regularly saved and backed up.

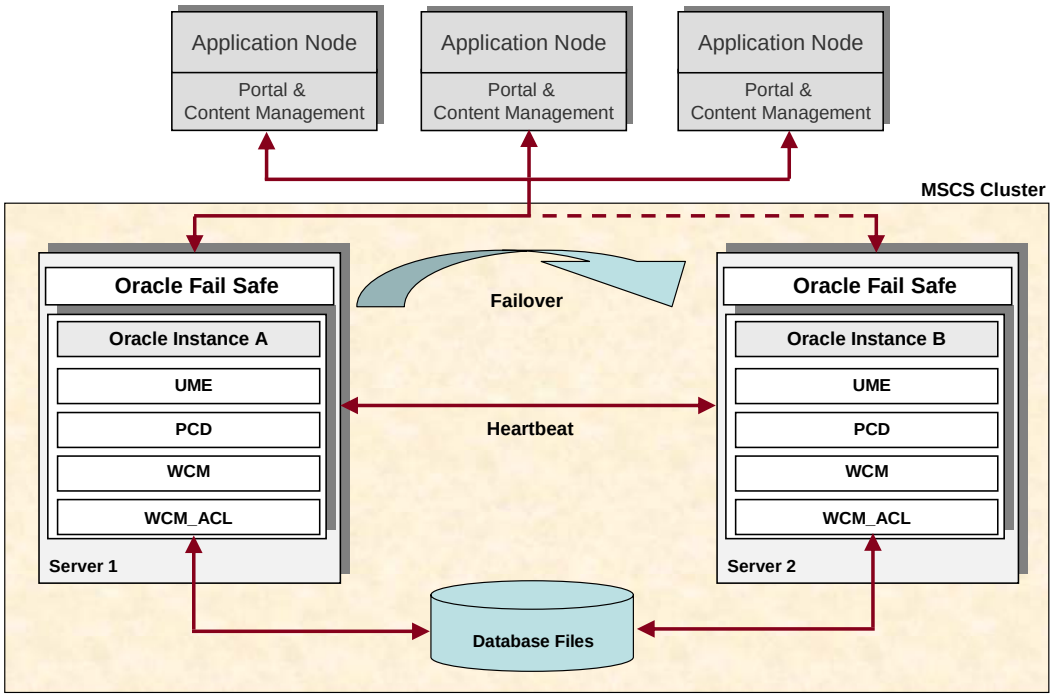
SAP Enterprise Portal can operate with active/passive database clustering so that if the active database fails, the second, passive, database can take over in a failover operation and continue to access the same shared resources. Clustered databases increase the availability of SAP components and overcome the single point of failure that an individual database represents.

Possible clustering solutions from third-party vendors include: HP MC/Service Guard (for Unix-based systems), Veritas clustering (for Sun Solaris, HP-UX or Windows), and Oracle Fail Safe with MSCS (Microsoft Cluster Server, for Windows NT).

Database clustering is not officially supported by SAP. Therefore, the customer must consult with the third-party vendor for recommendations and must assume all responsibility for guaranteeing the workability of such a solution.

Using third-party clustering solutions, transparent client failover can be performed with little or no user downtime, automatic reconnection of applications and users to the standby system and uninterrupted transfer of queries.

The following figure depicts a sample database cluster implemented through Oracle Failsafe in the MSCS cluster environment:



Rules to follow:

For the portal database:

- Determine high availability requirements by taking into account: number of active users, hours of operation, projected sizing, portal backup and restore plans, and special hardware requirements for the cluster, if any.
- If the Oracle Fail Safe solution is implemented, the database servers must be connected by Microsoft MSCS, so that the portal can access the database cluster as if it were a single “virtual” server. (Note that this is just one option among other third-party solutions.)

3.3.2 User Persistence Store

In order to ensure that the data residing in the user persistence store remains highly available and accessible to the portal server, a number of alternatives are available:

- An active-passive cluster can be implemented in a load-balanced LDAP directory service. For more information see the *Solution Management Guide,* accessed from SAP Service Marketplace at **`service.sap.com/ep60`** → *Documentation & More* → *Fundamentals* → *EP 6.0 SP2: Solution Management*.
- On HP UNIX servers, data should reside on RAID or mirrored disks using a Linux Network File System (NFS) server.

3.3.3 Central Configuration Data Directory

In a cluster environment, all the portals access central configuration data directory, located on a single physical server, in the folder, `/usr/sap/<sid>/global`. This directory stores configuration information for the PCD and Content Management (CM); therefore, each portal in the cluster must have direct access to this data.

In order to ensure cross-cluster access to and high availability of the central configuration data directory, it should be located on external RAID or mirrored disks, preferably part of the shared storage environment of the **portal database**.

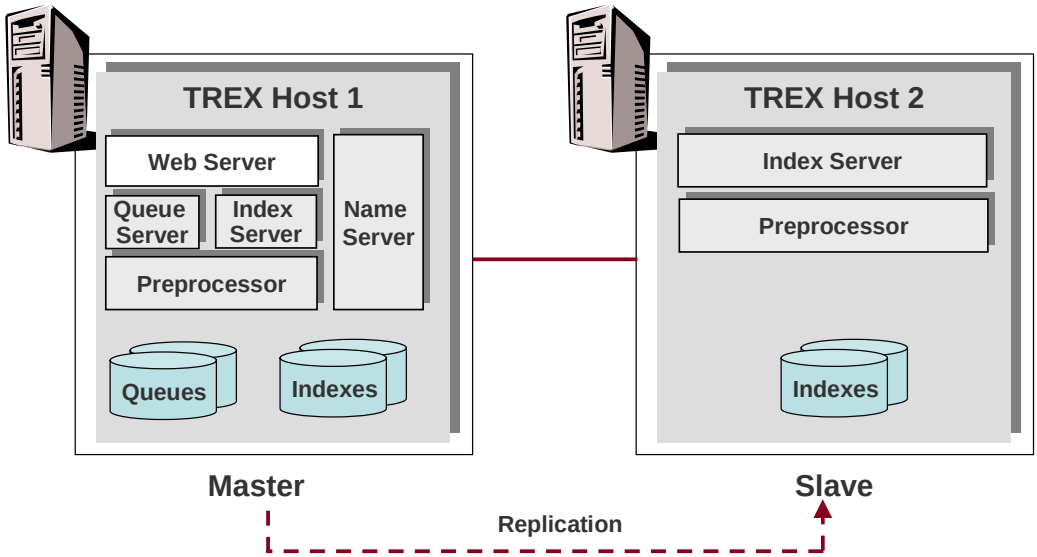
3.4 Search and Classification (TREX)

The TREX server is used in SAP Enterprise Portal to index, search and text-mine information requests received from the portal. For detailed information about this component, see the following documents:

- *SAP Enterprise Portal Technical Infrastructure Guide*
- TREX description, accessed from SAP Help Portal at **`help.sap.com/ep`** → *Administration, Developer and End User Documentation* → *Administration Guide* → *Knowledge Management* → *TREX Components*
- TREX feature list, accessed from SAP Service Marketplace at **`service.sap.com/ep-km`** → *Features and Presentations - KM with EP 6.0* → *Feature Lists* → *SAP EP 6.0: Overview Feature List (TREX).*

Search and Classification (TREX) should usually be deployed on a separate machine, running either UNIX or Windows.

TREX can be set up for high availability in a multiple-host master-slave configuration, as in the following figure:



The master host performs indexing, classification and searching, while the slave operates as a dedicated search server. It stores copies of the original indexes from the master host.

Due to the flexible architecture of TREX, the configuration can be scaled further, distributing the search and indexing load among several hosts and thereby ensuring the availability of the retrieval and classification functions. TREX scaling information can be found in the *Requirements and Recommendations for Using TREX* guide, located at SAP Service Marketplace at: **`http://service.sap.com/ep60`** → *SAP Enterprise Portal* → *Product Information* → *SAP Enterprise Portal 6.0* → *Documentation & More* → *How-To Guides* → *Current How-To Guides* → *TREX Recommendations.*

3.5 Unification

SAP Unification Server enables SAP Enterprise Portal users to relate business data across applications and/or databases. Although it runs within the portal landscape, the server has a discrete persistence layer consisting of a Database Unifier project repository, and its own load balancing services. It also includes a mechanism for installing additional unifiers for SAP applications such as BW and R/3. A full explanation of the architecture of the Unification Server is available from SAP Help Portal at **`help.sap.com/ep`** → *Administration, Developer and End User Documentation* → *Administration Guide* → *Cross Platform* → *Architecture of SAP Enterprise Portal* → *Portal Platform* → *Unification*.

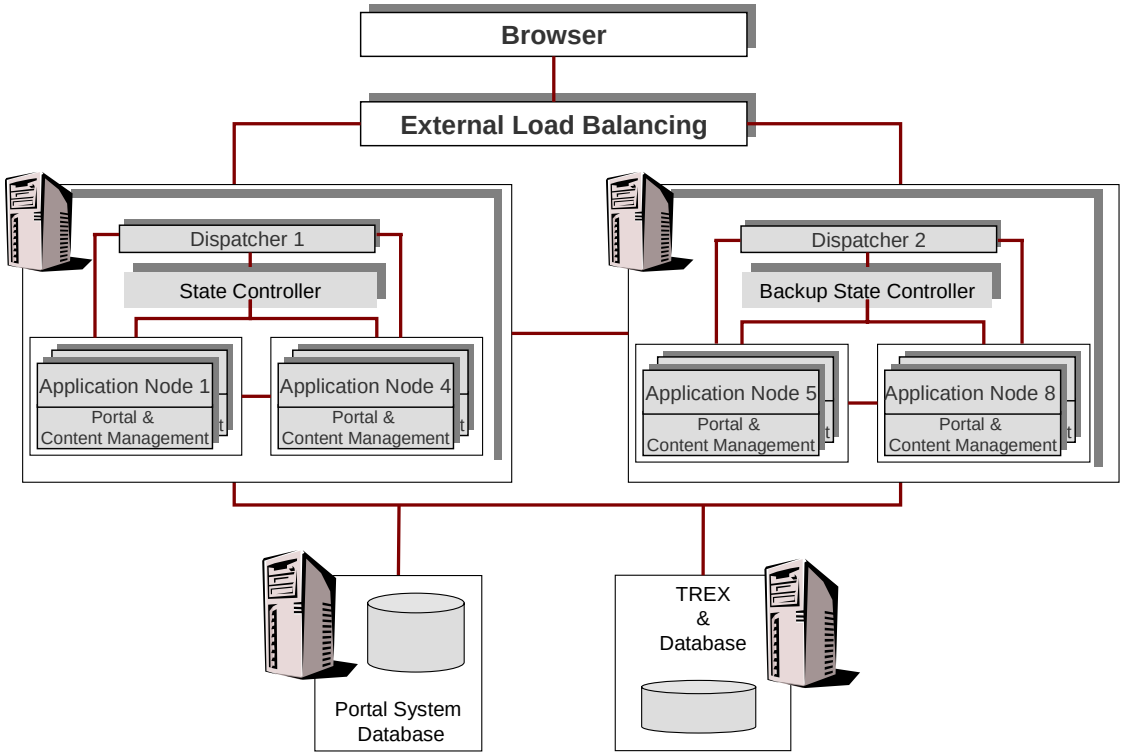
The Unification Server should be located on a machine that stands alone in the portal landscape. The server includes its own load-balancing mechanism, which runs in parallel to that of the portal. A special Unification load balancing tool needs to be run for each machine. For high availability purposes, the Unification Server can be replicated, and for each server – or Unification project – the Unification load balancing tool must be run. For full information, see *Unification Server Under Load Balancing,* an attachment to SAP Note 646094.

4 High Availability Scenarios

This section provides a number of different scenarios that could be implemented to produce an SAP Enterprise Portal of differing degrees of high availability. Selection of a scenario is based on the customer needs: number of active users, performance requirements, scaling plans and cost considerations. The scenarios are ordered from most basic to that which provides the highest degree of security and availability.

The Unification Server option is not included in these scenarios, but can be implemented as needed.

4.1 Highly Available SAP Enterprise Portal without Database Clustering

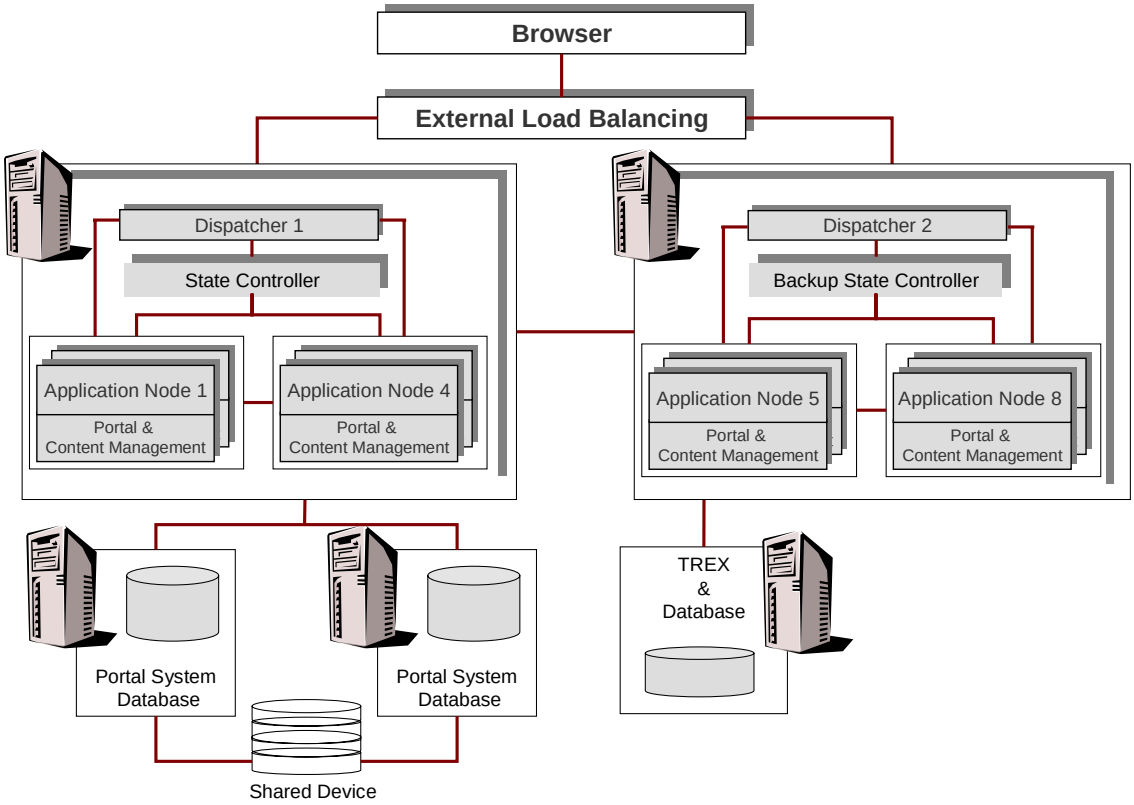


This scenario is a basic high-availability configuration. It uses a hardware load balancing mechanism for active/active portal clustering, with internal J2EE Engine clustering on each host. The database (in the persistence layer) and TREX are installed on separate machines. The load balancer is also replicated so that it does not become a single point of failure as well.

Remaining single points of failure:

- Persistence layer: Portal system database and user persistence store
- Search and Classification (TREX)
- External load balancer

4.2 Highly Available SAP Enterprise Portal with Database Clustering

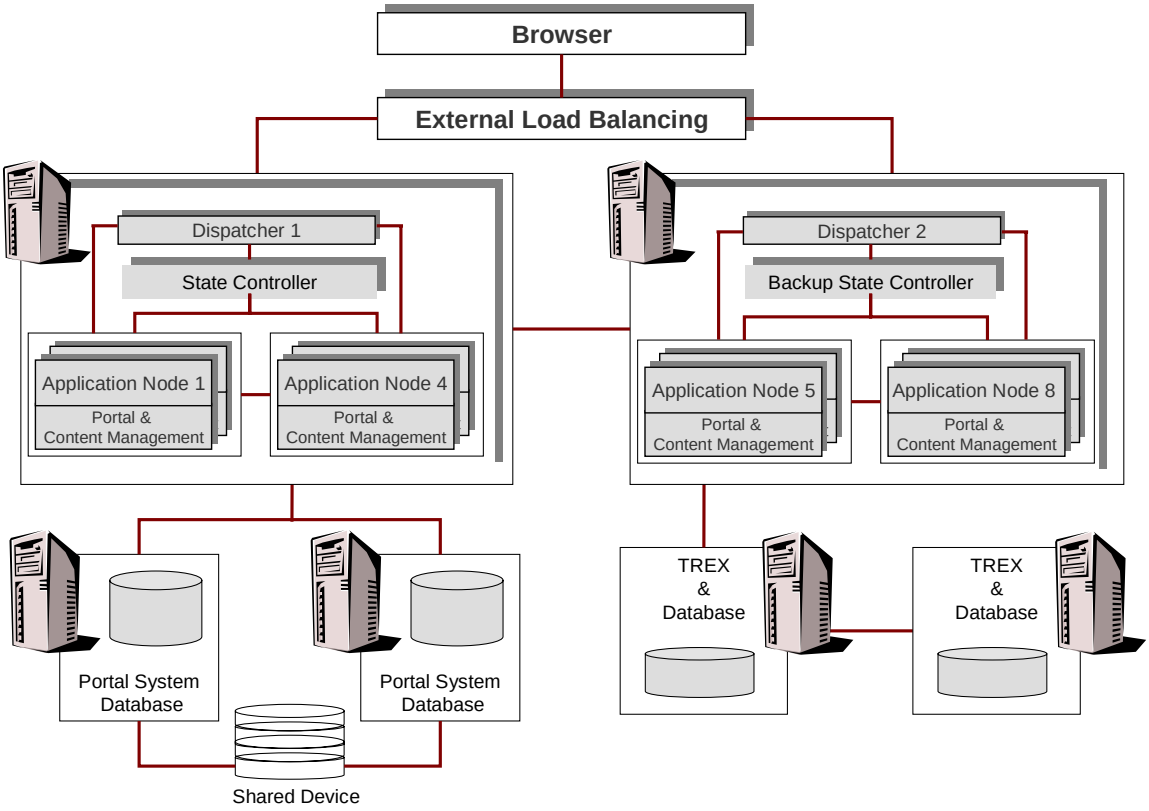


This scenario is the basic high-availability configuration with added clustering for database failover, in active/passive mode. The user persistence store can also be replicated here for high availability (but is not shown). TREX is installed on a separate machine.

Remaining single points of failure:

- Search and Classification (TREX)
- External load balancer

4.3 Highly Available SAP Enterprise Portal with Database and TREX Clustering

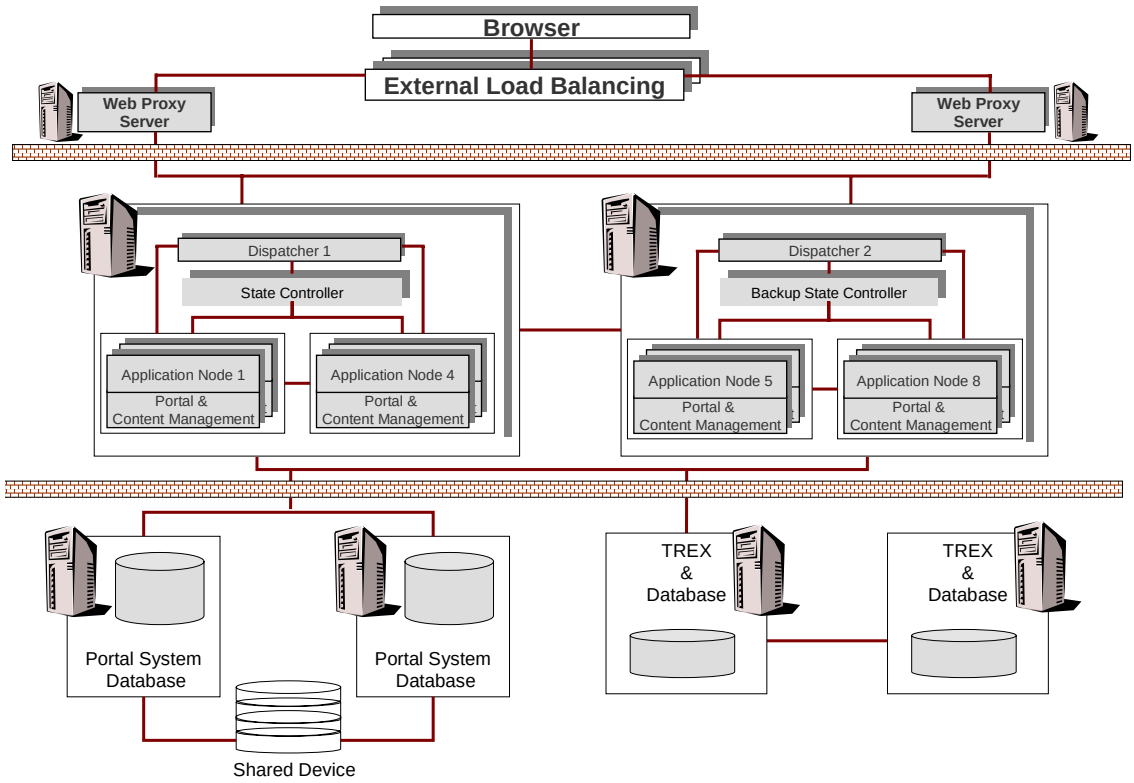


This scenario is a comprehensive configuration for high availability. Not only are the persistence layer and TREX separated from the portal cluster, but each unit is replicated for high availability as well.

Remaining single point of failure:

- External load balancer

4.4 Highly Available and Secure SAP Enterprise Portal



In this DMZ scenario, firewalls are implemented to protect the portal servers (at the top of the figure) and the persistence layer and TREX (at the bottom). The Web servers are proxy servers with the load balancer above them, so that the portal host connects to the proxy servers and not directly to the dispatchers. Note that each dispatcher of a J2EE Engine cluster requires its own Web server. The external load balancer has been replicated so that in this configuration, there should be no single point of failure.

For detailed information about building a network landscape for SAP Enterprise Portal, see the *SAP Enterprise Portal Technical Infrastructure Guide*. For more information about secure communications, see the “Communication Between Internal Components” section of the *SAP Enterprise Portal Security Guide*, accessed from SAP Service Marketplace at **`http://service.sap.com/ep60`** → *Documentation & More* → *Fundamentals* → *EP 6.0 SP2: Security Guide.*

5 Case Studies

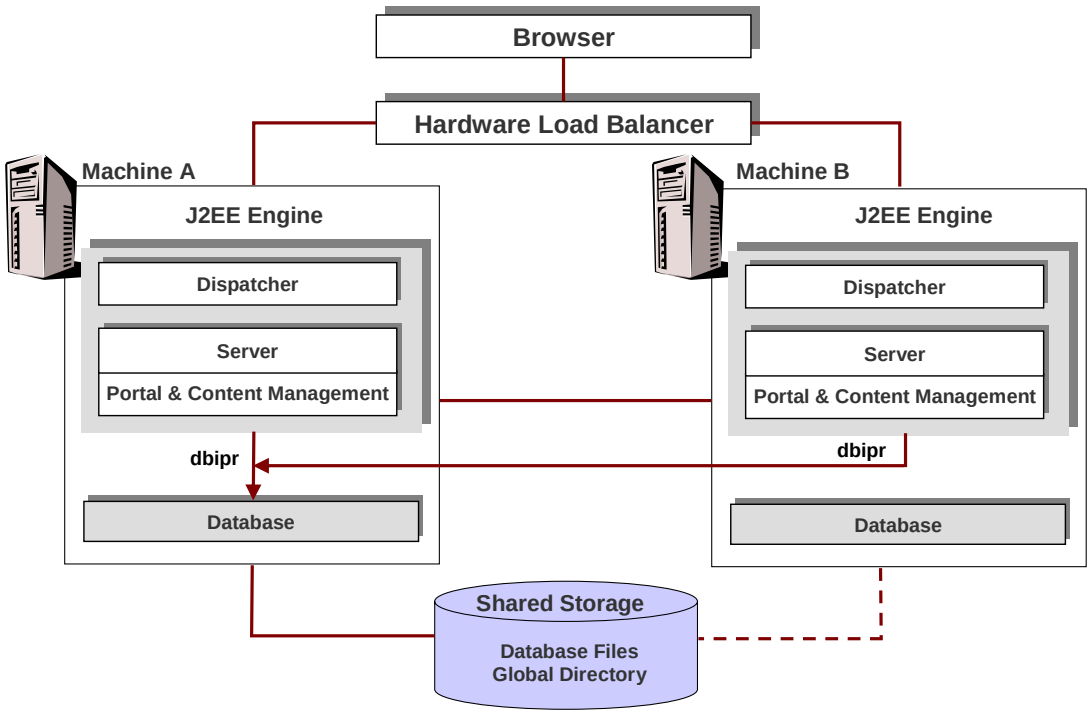
This section describes a number of different SAP Enterprise Portal installations, for which high-availability considerations have been implemented.

Notes:

- The case studies in this section include general information about specific third-party database clustering technologies. While these solutions have been implemented in either laboratory trials or actual customer environments, *SAP does not provide support for their implementation.*
- The case studies described here implement SAP Enterprise Portal running on J2EE Engine PL19, which does not operate with a state controller, as described in section 3.1, J2EE Engine.

5.1 HP-UX Internal Case Study

In this configuration, a hardware load balancer with active/active portal clustering is implemented between two machines, running on HP-UX. One machine includes both a J2EE-based SAP Enterprise Portal cluster as well as an Oracle database. The other machine runs a similar SAP Enterprise Portal cluster.



High availability is implemented in this scenario as follows:

- **Portal and SAP J2EE Clustering**
  Two machines have been clustered for the portal: Machine A runs the portal on J2EE Engine, as well as the portal database, running Oracle. Machine B runs another portal instance on J2EE Engine.

  The portal is accessed by the client through a common URL that points to a hardware load balancer. The load balancer distributes the incoming requests to the different SAP J2EE servers according to the distribution policy (such as round robin or fewest connections) defined for it. If any J2EE Engine is not operable for any reason, the load balancer detects the failure and routes all incoming requests to another engine. In order to enhance performance, an affinity based on client IP has been defined in the load balancing mechanism. Note that in this scenario, the hardware load balancer becomes a potential single point of failure and as such, requires a solution for high availability.

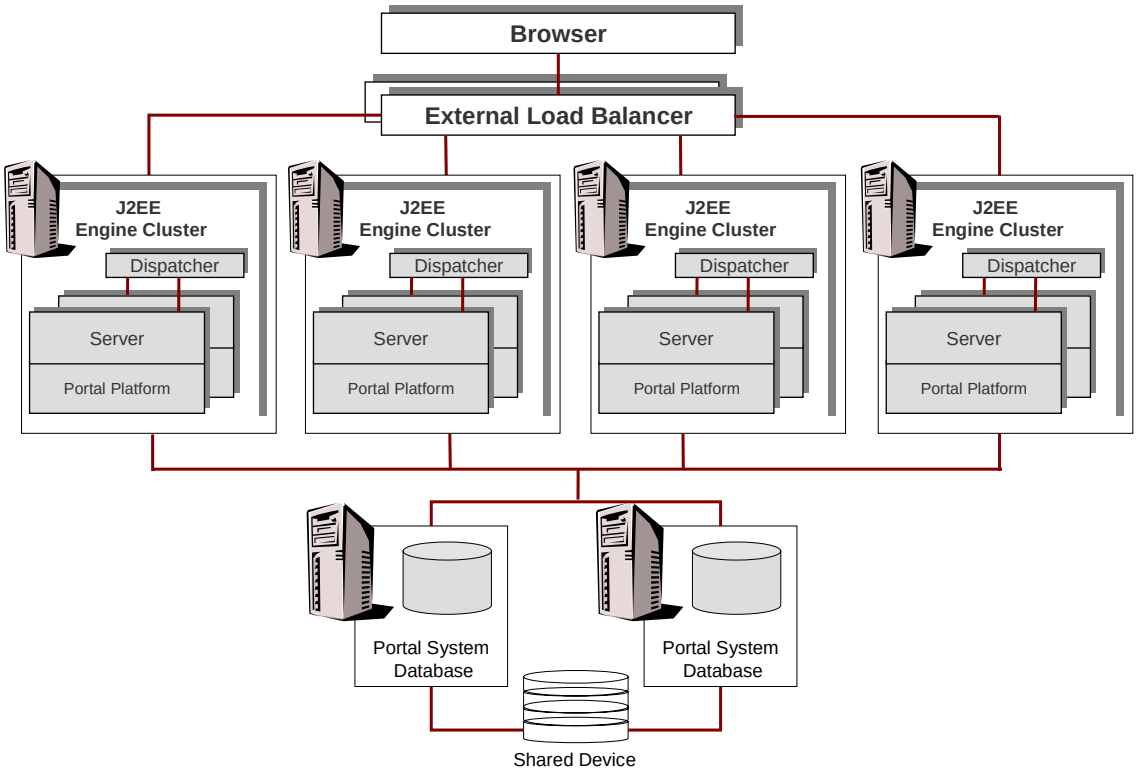
- **Database**
  In this scenario, the Oracle database runs on server A, in an active/passive cluster, implemented using the HP MC/ServiceGuard cluster product. The database is accessed by the J2EE Engine on that machine using the relocatable IP address dbipr. If server A encounters a hardware failure, the clustering software relocates the package – including its shared resources (disk and IP address) – and automatically starts the application on the database located on server B. These resources can move (relocate) with the package from one cluster node to another if the first node fails for some reason.

- **Persistence Store**
  In order to ensure that all servers of the portal landscape can access the Central Configuration Data Directory and CM, an external disk is used for the central storage, within the persistence store. The disk is secured from failure by use of NFS, which is made highly available by a cluster product. In order to prevent the disk itself from representing a single point of failure, the data actually resides on mirrored disks (alternatively, a RAID configuration could be used).

5.2 International Company A

This section describes a standard SAP Enterprise Portal installation, configured for high availability.

This configuration includes four J2EE Engine clusters, two server nodes on each machine. The portal cluster is load balanced using a redundant load balancer for high availability. The database resides separately on two replicated machines. There is no CM or TREX.

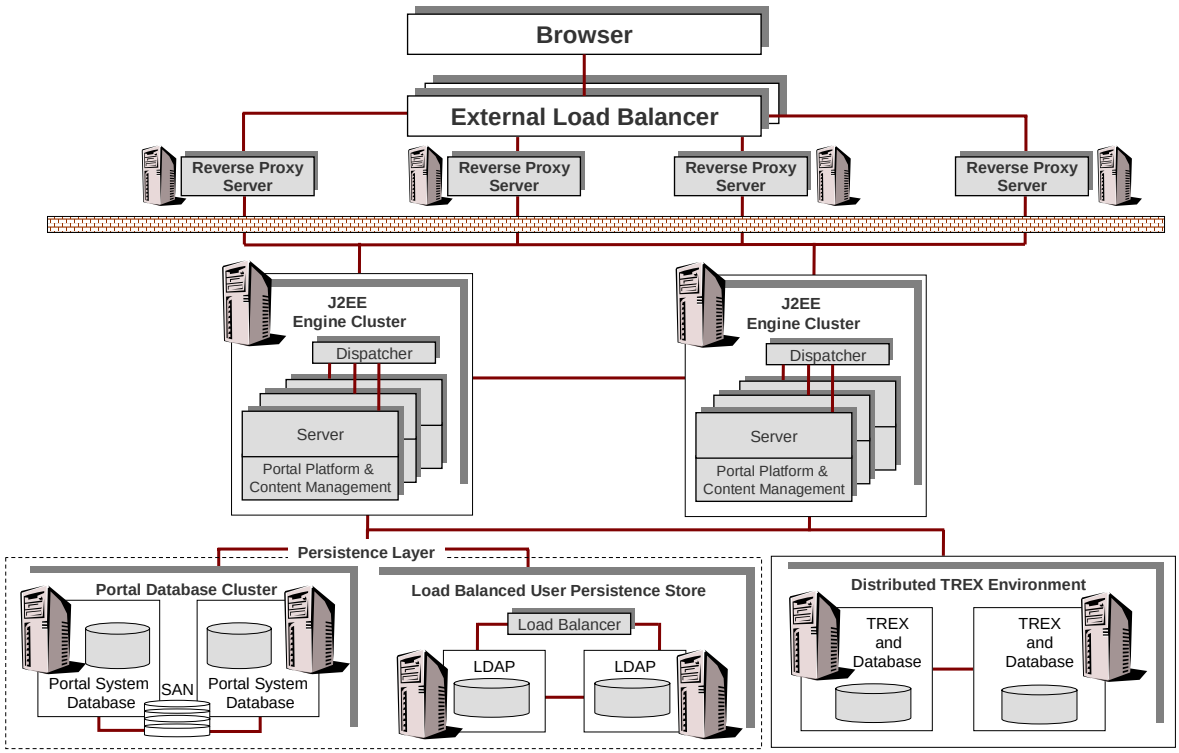


High availability is implemented in this scenario as follows:

- **J2EE Engine Clustering**
  Four machines, running Windows, have been clustered for the portal: Each machine runs a J2EE Engine cluster, with a single dispatcher and two server nodes.
- **Portal Clustering**
  The portal is accessed by the client through a common URL that points to an IBM SecureWay Network Dispatcher, the load-balancing component of IBM WebSphere Performance Pack. The load balancer distributes the incoming requests to the J2EE server that is most available when the request is received. In order to avoid bottlenecks and guarantee high availability, the load balancer has been replicated. The WebSphere dispatcher was selected because it can dispatch client requests to the J2EE servers, which in turn can send the requests directly back to the clients without having to go through the dispatcher.
- **Database**
  In this scenario, an Oracle database runs on two Linux-based servers in a separate active/active cluster, accessible by all J2EE Engines in the cluster. (The use of Linux platform is exceptional here, and not a recommended SAP standard.)

5.3 International Company B

The configuration described here is a large, secure installation that includes a complementary QA environment. An array of reverse proxy servers are used between the load balancer and the portal cluster to ensure a secure environment. Clustering or load balancing have been implemented at all levels – including TREX and the user persistence store – to ensure high availability at all potential points of failure.



High availability is implemented in this scenario as follows:

- **J2EE Engine Clustering**
  Three Sun Fire servers have been clustered for the portal: Two machines – each running three instances of J2EE Engine – are used for the portal production environment.
- **Portal Clustering**
  A load-balanced, clustered portal environment is created using hardware load balancing and Novell iChain reverse proxy servers, which provide security by preventing direct access to the portal. In this case, the proxy servers serve as agents between the load balancer and the portal, transmitting the client requests coming through the load balancer to the portal servers and retransmitting the content back to the portal clients.

- **Database**
  In this scenario, active-active database clustering is achieved for the Sun Fire servers running Oracle. The disk is made highly available using the Veritas database cluster software. Both servers share the database files stored on a storage array network.
- **Persistence Store**
  Also in the persistence layer in this configuration is the user persistence store, organized in a load-balanced LDAP configuration.
- **Search and Classification (TREX)**
  Two Sun Fire machines in a master-slave environment constitute a distributed TREX.