SAP Enterprise Portal Technical Infrastructure

SAP Enterprise Portal 6.0 SP2

Document version 2.0

Copyright

Icons in Body Text

Icon	Meaning
⚠	Caution
💬	Example
💡	Note
🧭	Recommendation
SYN	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 the any version of *SAP Library*.

Typographic Conventions

Type Style	Description
Example text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
`Example text`	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE 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.
`Example 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.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

Contents

SAP Enterprise Portal Technical Infrastructure

Purpose

This guide provides a brief overview of the architecture of SAP Enterprise Portal, its platforms and components. It describes how to distribute the components of the platforms, and the possible deployment landscapes that ensure high availability, and optimized performance of the portal processes.

High availability is the ability to deploy and implement a landscape that ensures continuous operation of the portal. It includes the capacity to identify points-of-failure in the system that can disrupt end users' work, and the ability to remove the identified points-of-failure.

Explanations on how to make the portal highly available include discussions on clustering and load-balancing methods and mechanisms.

In addition, this guide explains possible scalable techniques that can be implemented to optimize the performance of the portal in a highly available environment. Scalability refers to ways of distributing portal components to handle increased loads as demand for portal resources grow.

Security is a primary concern for enterprises as the portal becomes highly available. This guide also describes deployment and implementation of the portal in a secure network landscape.

The guide does not provide details of all the possible ways of configuring the portal in a network infrastructure, as there are several different ways and utilities that enable portal deployment and configuration.

Implementation Considerations

Partners, consultants, and implementers who need to provide highly available, scalable performance and a reliable failover solution can use this guide.

The approach discussed in this guide does not replace the information for installing SAP Enterprise Portal; rather it should be used as supplement to the SAP Enterprise Portal installation guides.

Overview of SAP Enterprise Portal

SAP Enterprise Portal is comprised of the following:

- Portal platform
- Knowledge Management platform
- Collaboration

Each of the above programs consists of SAP technologies that can run other processes as well.

Portal Platform

The portal platform is integrated into SAP J2EE Engine, an SAP proprietary application server based on the Java 2 Enterprise Edition (J2EE ™) standards. In addition to providing Java-based enterprise resources, the application server provides an HTTP servlet Web server to the portal.

The system that implements security mechanisms for authentication and authorization, and allows access to various content, including single sign-on (SSO) for users, is the User Management Engine (UME), an SAP proprietary mechanism that is integrated into the portal.

The portal platform consists of the following:

- Portal framework

  The portal framework is a virtual environment that enables the portal both as a development and a runtime environment.

  Applications that run in the portal, such as iViews, are implemented by portal components and portal services including the Portal Content Directory (PCD). The following components are required in a landscape of the portal:

  - Portal System Database, a relational database management system (RDBMS) to hold various data and metadata required by the portal at run time. The information in the database is used by the portal and the Knowledge Management platforms.
  - User Persistence Store, a storage area containing information about users, such as directory servers.
  - Java Development Kit (JDK), the Java runtime environment required by the Java software and components in the portal.

- Unification

  Unification consists of the following:

  - Microsoft® Internet Information Server (IIS)
  - SAP Unification Server
  - SAP Unifiers

- Connector framework

  The connector framework is comprised of adapters that provide connectivity between an Enterprise Information System (EIS) and the portal.

Knowledge Management

The Knowledge Management platform is comprised of:

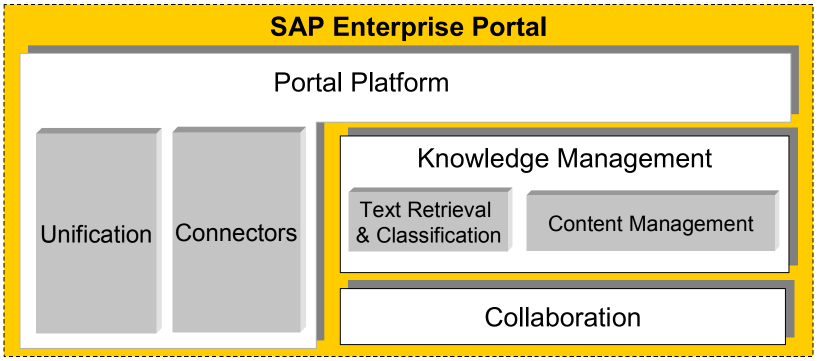
- Content Management (CM)
- Text Retrieval and Classification (TREX)

Collaboration

Collaboration consists of the following:

- Collaboration Launch Pad
- Collaboration Rooms
- Real-Time Collaboration
- Asynchronous Collaboration
- Groupware Framework
- Synchronous Collaboration Framework

The following illustration shows the platforms of SAP Enterprise Portal:



Access to the components of all the platforms in SAP Enterprise Portal is via the portal platform.

Identifying Portal Resources at Run Time

Categorizing the resources required when the portal loads, lets you identify those resources that are common to the platforms of SAP Enterprise Portal and can be shared, from those that should be distributed over several physical machines.

Portal resources are of two types; runtime resources, and the storage resources used during runtime.

Runtime resources consist of portal components and services that assemble portal pages, iViews, manage themes and styles. These resources use stored data in the database system and in the user persistence store to implement the user interface of the portal.

The following resources of the portal are needed at runtime:

- SAP J2EE Engine

  As an integral part of the portal platform, every deployment of the portal requires an instance of the SAP J2EE Engine.

  Highly available and a scalable portal infrastructure can be implemented using the loading-balancing, and clustering mechanisms of SAP J2EE Engine.

- Portal Framework

  The portal framework provides the working environment for the portal. It allows a user to work as a portal administrator, a developer, or an end user, depending on the user's logon profile.

  Inter-operation and integration between the components and services that implement the portal infrastructure and functionality exist in this environment.

The storage resources maintain stored data used at runtime. These resources provide common applicable functionality to the platforms of SAP Enterprise Portal.

The storage resources include the following:

- Portal System Database

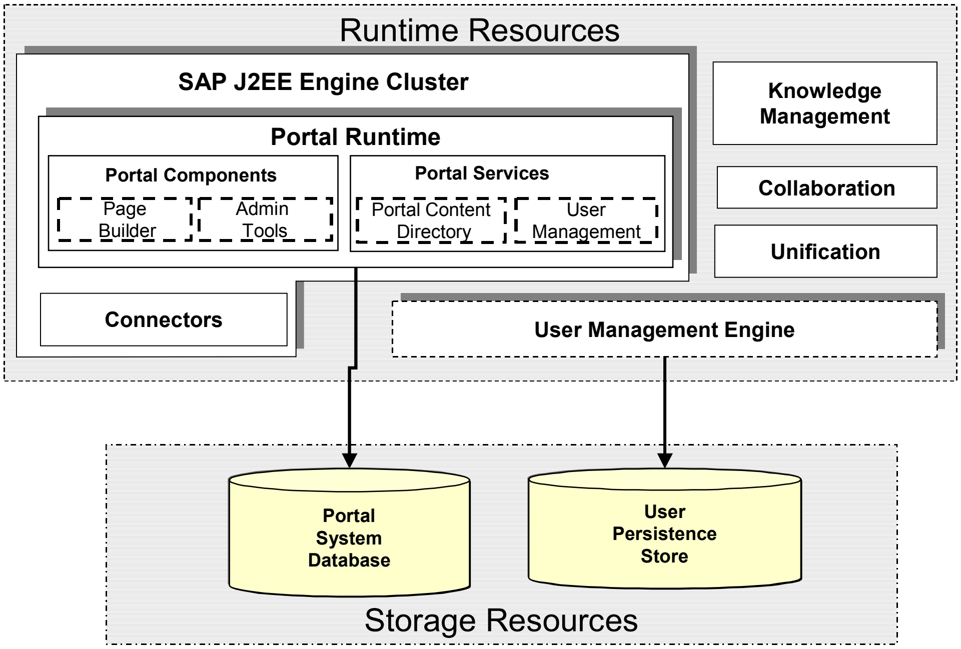
  This is the database for SAP Enterprise Portal, which is usually located in a separate machine. Database schemas for the Portal Content Directory (PCD), User Management Engine (UME), and Content Management (CM) are stored in this database.

  In addition, the database stores portal runtime objects, including role definitions, and page-to-role relationships, deployable portal archive (PAR) files, templates, personalization data, and it is the source of several other objects.

- User Persistence Store

  Refers to user data stored in one or more repositories. User-related data repository might be a database, a Lightweight Directory Access Protocol (LDAP) directory, and SAP R/3 System.

The following illustration shows the resources of the platforms that loads at runtime:



See also:

Distributing the Components of SAP Enterprise Portal [Page 12]

Planning the SAP Enterprise Portal Cluster [Page 18]

Planning the Portal Landscape

The portal landscape defines a view of the portal infrastructure at a point in time. You can deploy the portal in stages, however, each stage has an implementation plan that is distinct to the landscape. The following stages help to plan and implement the portal landscape:

1. Development

   The development stage consists of installation of the portal platform for content development.

2. Integration and Testing

   Integration and testing involves fine-tuning and troubleshooting issues such as, validating installations and testing the content developed during the development stage.

3. Production

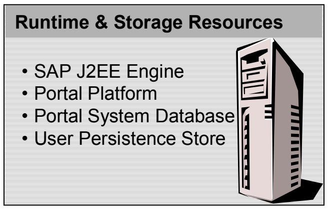
   The production landscape refers to the capability to scale the portal and to implement high availability solutions after completing the development stage. This environment includes additional portal installations in a cluster.

As you deploy the portal in stages, any change implemented after completing the production stage, such as applying upgrades, support packages, patches, and other applications changes the environment. It is recommended that, you first test the changed environment before deploying the change in the production environment.

Development Stage

The development stage can consist of one or two servers. This usually depends on the number of users and the amount of content.

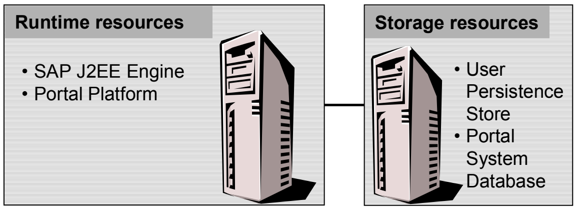
Technical Landscape: Deploying the portal for content development in a single server



Although a landscape for the development stage can consist of a single physical server, it is recommended to deploy the portal in two physical servers; one server to run the portal runtime resources, such as SAP J2EE Engine, and the other to maintain the storage resources required at runtime, such as the portal system database.

The following is an illustration of the components installed for development landscape:

Technical Landscape: Deploying the portal for content development



At the development stage, the primary focus is to create content (iViews) for various users. Two kinds of portal users will be working in this landscape, portal administrators and content developers. Issues facing portal implementers are content-related and not availability and performance of the portal.

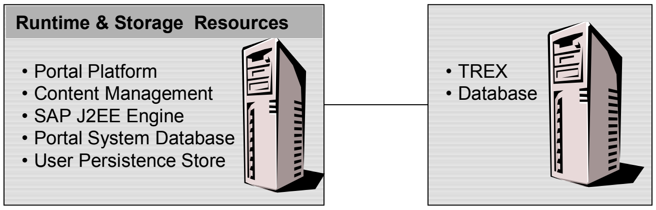
Integration and Testing

Integration and testing is a continuous process that must take place in any landscape. It follows the development stage, such as changes that are introduced after the initial portal deployment.

For instance, when you install Content Management in addition to the portal platform, you need to test both components before expanding the portal environment for production.

The illustration below is an example of an integrated portal landscape:

Technical Landscape: Deploying the portal for integration and testing



Portal administrators play an active role in this stage, such as assigning iViews to users, and ensuring that users have the proper security credentials to content of iViews assigned to them.

Production Stage

The production stage is an extension of the development stage. It begins when content development, and integration and testing have been successfully completed.

The landscape implementation in a production environment usually consists of two or more physical servers. The portal system database, and the user persistence store are placed separately from the portal platform installation.

In this stage, portal implementation can include several deployment cycles, where each cycle focuses on the deployment and implementation of specific SAP Enterprise Portal components, such as content management, collaboration, business packages and connectors.

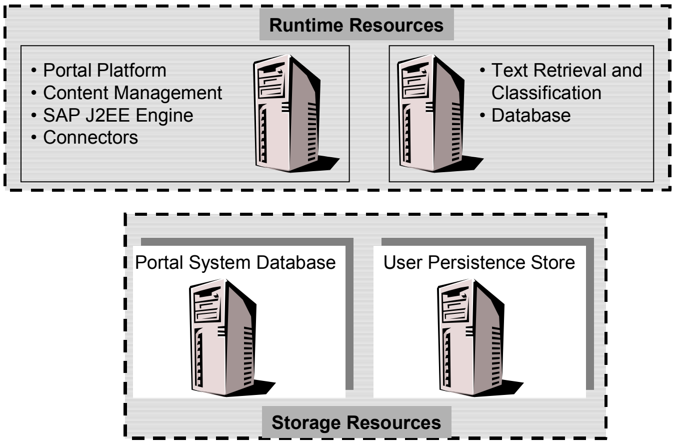
TREX, which is both a processor, and network intensive application (depending on whether or not it is indexing remote shares), must have a separate server.

Planning the production environment involves several experts including portal implementers, system administrators, database experts, and network security experts.

The challenging issues in the production stage relate to high availability, scalable landscapes, and optimized performance.

The following is an illustration of a landscape in the production environment showing runtime resources and storage resources:

Landscape: Deploying the portal in a production environment



See also:

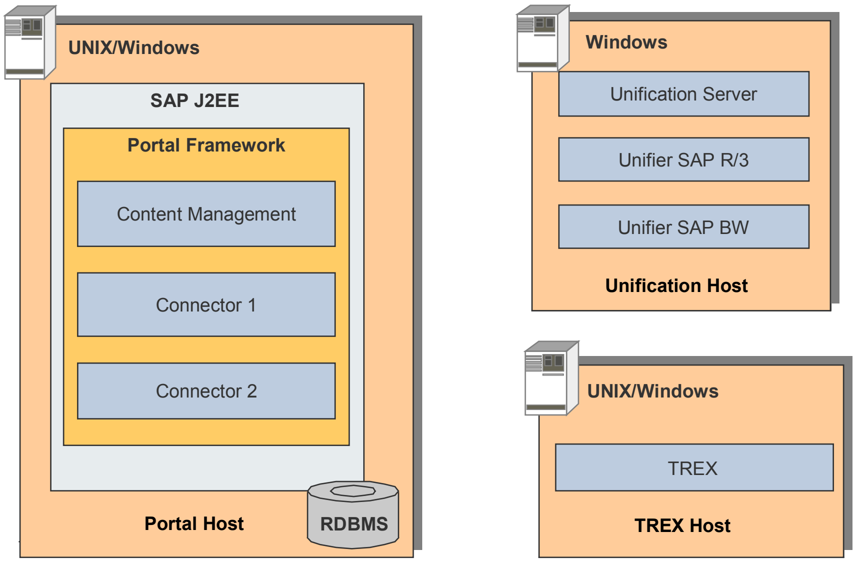
Load Balancing the Portal Cluster [Page 16]

Clustering the Persistence Layer [Page 15]

Network Landscapes [Page 47]

Distributing the Components of SAP Enterprise Portal

The infrastructure of the portal allows you to deploy the portal resources separately. The following illustration shows deployed components of SAP Enterprise Portal for development and testing:

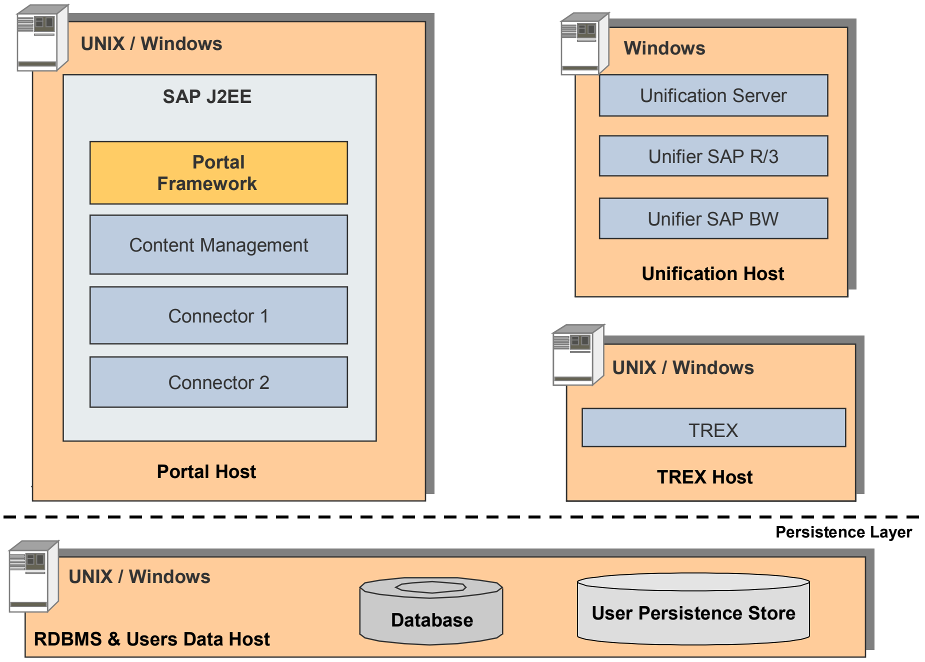


The portal host in the illustration above includes portal resources, such as the portal system database (RDBMS), and the user persistence store (user-related data) on a single machine. SAP Unification Server and SAP Unifiers running on Windows, are deployed on another machine. Content Management is always deployed on the portal host, however, it is recommended to deploy Text Retrieval and Classification (TREX) on a separate machine.

Setting Up a Persistence Layer in a Separate Machine

An alternative method of deploying SAP Enterprise Portal for development and testing is to locate the portal system database, and the user persistence store on a machine other than the portal host.

The following illustrates the alternative deployment method of SAP Enterprise Portal:



When a single portal host has to serve many portal clients, as in the above landscape, demand on the resources and processes can become an issue.

You can add more servers to host additional portals, forming a cluster of portals. Implementing hardware, or software load-balancing solution for the portal cluster makes it highly available and accessible to a large number of users, or portal clients.

See also:

Clustering the Porta [Page 14]l

Planning the SAP Enterprise Portal Cluster [Page 18]

Preparing the SAP Enterprise Portal Cluster [Page 28]

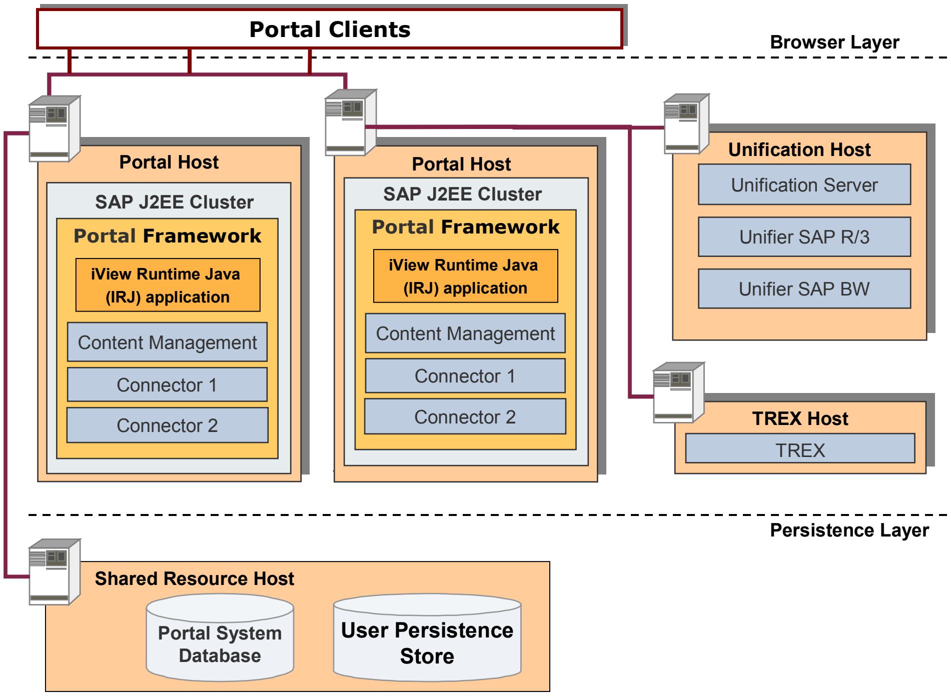
Clustering the Portal Platform

Purpose

In networking infrastructure, a cluster is the combination of two or more servers configured to run a common set of applications and resources as though they are on a single system.

For a cluster of portals, you configure the SAP J2EE Engine in order to deploy the portal runtime resources across several physical machines. SAP J2EE Engine enables a cluster of portals to simultaneously connect to each other, and to access a single persistence layer in which the storage resources required at runtime are shared.

The following illustration shows distributed portals accessing a single persistence layer:



See also:

Planning the Portal Cluster [Page 18]

Clustering SAP Enterprise Portal [Page 29]

Clustering the Persistence Layer

Implementing the persistence layer in a cluster makes the central storage, or shared resources, highly available and reliable. Also, when a failure occurs in the persistence layer, resources can be redirected and the workload redistributed to other machines.

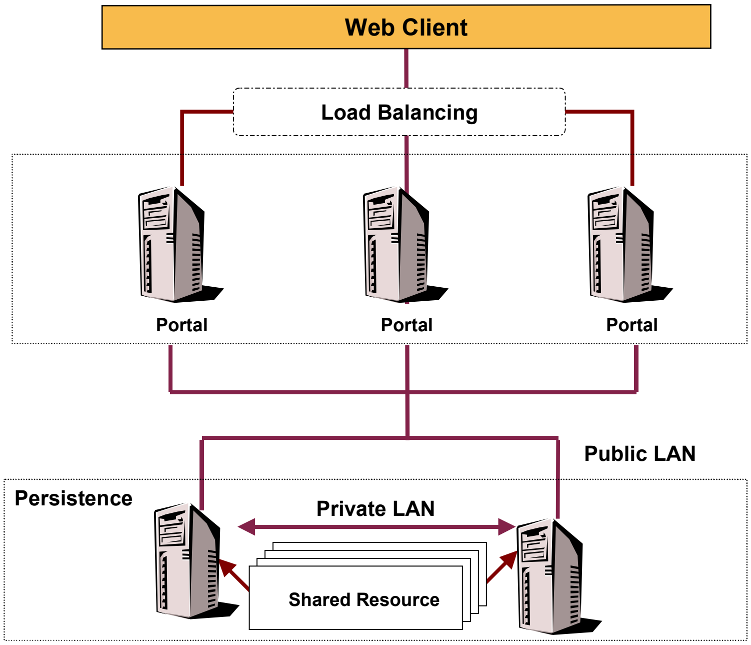
Clustering solutions are generally vendor-specific. In your implementation, you need to address the issues of clustering, and involve systems administrators, database system (RDBMS) experts, and Web server administrators.

To identify scalable performance and high availability needs for the portal at the customer's site, determine the following:

- Special hardware requirements of the clustering solution
- Number of users
- Hours of operation
- Projected changes in size and performance requirements
- Portal backup and restore plans

The following illustration shows the clustered persistence layer and a load-balancing environment:

Clustering the Persistence Layer



See also:

[Load-Balancing the Portal Cluster [Page 16]]

Load Balancing the Portal Cluster

Purpose

Load balancing is the ability to distribute the work that one server has to do between two or more physical machines, so that more work gets done in the same amount of time, and users are served faster. Implementing a load balancing solution requires multiple servers, and allows you to apply failover and backup capabilities.

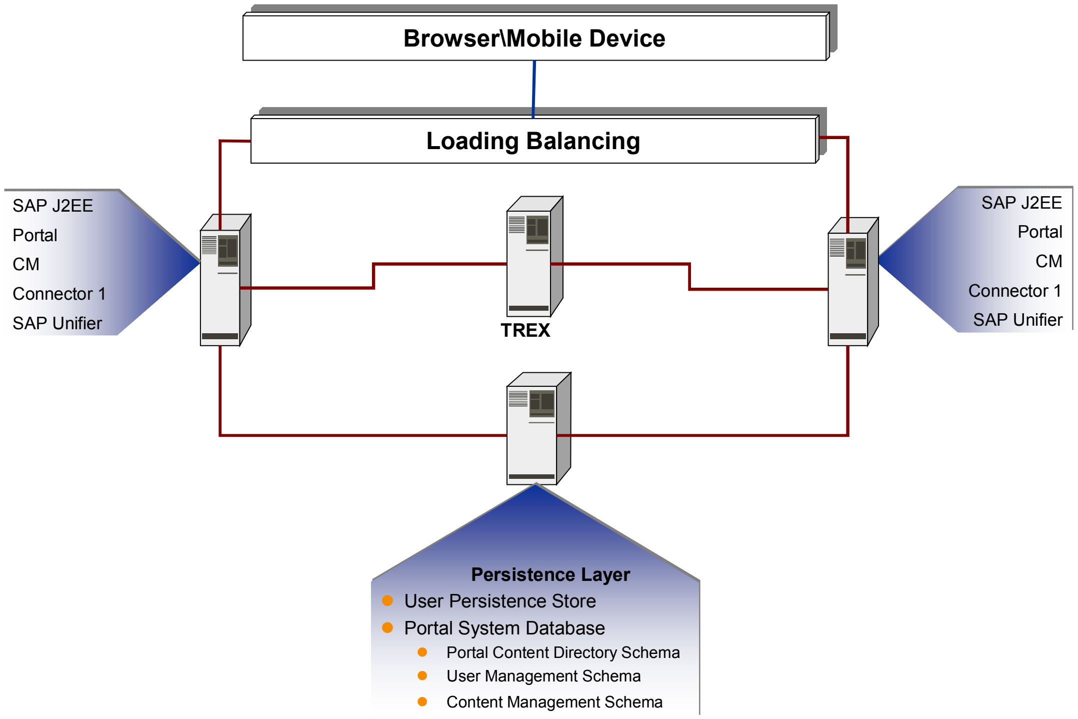
You can load-balance distributed portal runtime resources in a cluster with hardware, or software mechanism, or a combination of both.

For load balancing the portal cluster, there are several approaches; one approach is to use an inherent load balancing mechanism in SAP J2EE Engine to route each HTTP request intended for the portal cluster to a different server.

For additional information on load balancing using SAP J2EE Engine, refer to SAP J2EE Load Balancing Mechanism. [Page 26]

Where two or more physical servers are used to balance a work load, a third server is needed to distribute the work load in the cluster.

The following illustrates the load balancing infrastructure in the cluster environment:



Most load-balancing solutions can be configured to have “stickiness” or affinity, that is, forcing all the requests of a specific client to be forwarded to the same server. In such a case, load balancing takes place only for the initial logon, thereafter, the client’s requests are directed to the selected server.

Caching and Synchronization

During runtime, each portal installation in the cluster environment creates a local temporary storage or cache to hold runtime objects. These objects come from the Portal Content Directory (PCD). Caching the content of the PCD improves performance by reducing the load on each portal and the demand on the resources used by the network.

Changes made by administrators such as personalized views or customized themes to any portal in the cluster, are stored in the PCD. Since the PCD is in the shared location (persistence layer) of the load balancing environment, such changes must be available to all the other portals.

Specific services immediately activate internal notification of changes to all the portals in the cluster. Each portal then synchronizes with the PCD in order to update its local cache. The synchronization mechanism also checks the PCD intermittently for changes in the list of objects in the PCD cache.

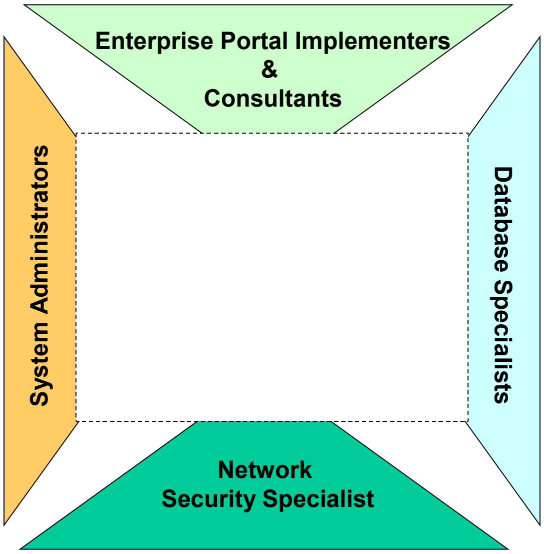
This mechanism enables the distributed portals to have a consistent view of the persistence layer.

Planning the Portal Cluster

When you plan to deploy the portal on several machines to form a cluster, you need to consider the possible load on the network infrastructure.

Planning a cluster for the portal is a joint effort among several specialists, like security experts, systems administrators, database specialists, Enterprise Portal implementers and consultants.

The following illustrates the areas that are affected:



Planning can be in stages:

1. Enterprise Portal implementers, consultants, security experts, and database specialists decide the general and customer-specific network requirements.
2. Enterprise Portal implementers, system administrators, security experts and database specialists work to meet the defined requirements, that may consist of the following:
    - Network topology and network components (routers, hubs, switches, servers, if necessary with multiple network connections)
    - Network configuration of the servers (IP addresses, routes, host names)
    - Installation and configuration of SAP J2EE Engine, Enterprise Portal, Content Management, Collaboration and TREX
    - Testing the connections and conditions under full network load.

When planning the portal cluster environment, consider installing the portal system database, and the user persistence store (storage for user-related data) separately, each on a dedicated server.

Scalability and Availability Considerations

You can scale the portal either by adding extra machines, or increasing specific components of SAP J2EE Engine in a single machine. Either way, you can implement a load-balancing solution that enables the portal to handle a large number of users.

You can implement an external load-balancing solution; that consists of load balancing solutions of third-party hardware or software vendors.

However, note that some load-balancing solutions have limitations on the number of servers in the cluster environment.

Alternatively, you can implement the load-balancing mechanism in SAP J2EE Engine for the cluster of portals.

For additional information on the SAP J2EE Engine, refer to the following sections:

- Managing Resources Through SAP J2EE Engine [Page 20]
- Planning SAP Enterprise Portal Cluster [Page 21]
- Performance and Availability [Page 24]
- Planning the Cluster Configuration [Page 21].

In a cluster environment, you need an additional installation of the portal on each machine. Additional instances of SAP Enterprise Portal is installed and configured for you, using SAPinst.

Front-End Devices

Issues relating to front-end devices for SAP Enterprise Portal include the following:

- Stable TCP/IP connection to front-end devices, such as browsers, and mobile devices
- Fine-tuning of front-end devices to enhance performance and the effects of upgraded installations

For detailed information on configuring your browser to optimize performance of the portal, refer to *How to Fine-Tune the Performance of SAP Enterprise Portal 6.0*, at: **service.sap.com/ep60,** `Enterprise Portal 6.0 > Documentation & More > How-To guides`

Backup and Restore Strategies

In a production environment, consider a backup and restore strategy. Strategies may include both offline and online backups.

If the data is backed up from a central point over the network, there is a heavy network load. If the backup occurs during operational hours of the portal (which includes background processing), consider the effect of increases in the delay time and limitations on the bandwidth that can affect the overall performance.

For detailed information on backup strategies for SAP Enterprise Portal, refer to the section on *Backup and Recovery* in the *EP6 Solution Management Guide*. Locate the guide at: **`service.sap.com/ep60,`** `Enterprise Portal 6.0 > Documentation & More > Fundamentals > Solution Management.`

See also:

Preparing SAP Enterprise Portal Cluster [Page 28]

Managing Resources Through SAP J2EE Engine

SAP J2EE Engine enables communication among distributed SAP Enterprise Portal installations over several machines in the network infrastructure. It manages and runs a set of portal servers and their shared resources as though they are on a single machine.

Managing the portal cluster environment is accomplished through two software components of SAP J2EE Engine, **dispatcher** and **server** nodes.

Together, the dispatcher and server nodes enable clustering, load-balancing, and fast response to client requests. Connection to clients is transparent, that is, a client has no direct connection to a server node in the cluster; rather clients communicate with the dispatcher nodes.

Dispatcher nodes are responsible for communication on two levels. First, they enable communication among all the server nodes; secondly, they provide communication between clients and other dispatcher nodes in the SAP J2EE Engine environment.

Server nodes process client requests, returning the responses to the dispatchers, which in turn redirect them to clients. Server nodes connect to one another and then to the dispatcher nodes in the SAP J2EE Engine environment. Server nodes host the business logic that complies with J2EE standards.

For detailed information on SAP J2EE Engine, refer to the section *Getting Started* in the documentation installed with SAP J2EE Engine, in the path:

<J2EE_location>\j2ee\j2ee_**<installation_instance>**\docs\index.html

See also:

Planning the Cluster Configuration [Page 21]

Performance and Availability [Page 24]

Planning the Cluster Configuration

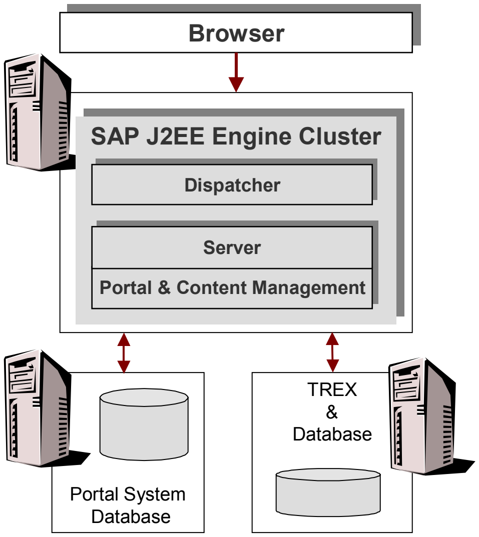
Two main factors that determine the configuration of SAP J2EE Engine environment are:

- **Number of portals to deploy**: The number of server nodes should be in relation to the number of portals to be deployed in the cluster.
- **Number of client requests**: The number of dispatcher nodes must increase in proportion to the increase in client requests.

The following are some configurations of the SAP J2EE Engine environment:

- Single server node, single dispatcher node

  This is the default configuration of SAP J2EE Engine, where the load factor and request processing of the single server node is not an issue.



- Single dispatcher node, with multiple server nodes

  In a single machine, you can configure several server nodes to communicate with a single dispatcher. Doing so allows you to optimize system resources used for SAP J2EE Engine processes.

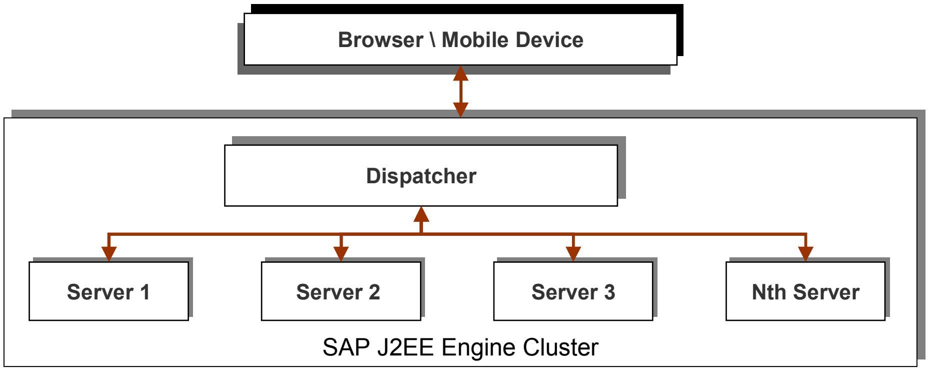
  Specifically, you set the value of the property, *LocalLoadBalancing,* to TRUE in the SAP J2EE ConfigTool window. In this configuration, the dispatcher communicates only locally with the servers running on the same host.

  The inherent advantage is that connections within the local host are faster and more efficient. All client requests are sent to the dispatcher node, which then redirects them to the server with the least load, for processing.

  When the number of requests increases, extra server nodes can be added to achieve greater scalability, and to reduce response time. This configuration enables optimal load-balancing because the main use of the dispatcher is to reroute client requests.

The following figure shows a single dispatcher node with several server nodes:

Optimizing SAP J2EE Engine Cluster on a single machine

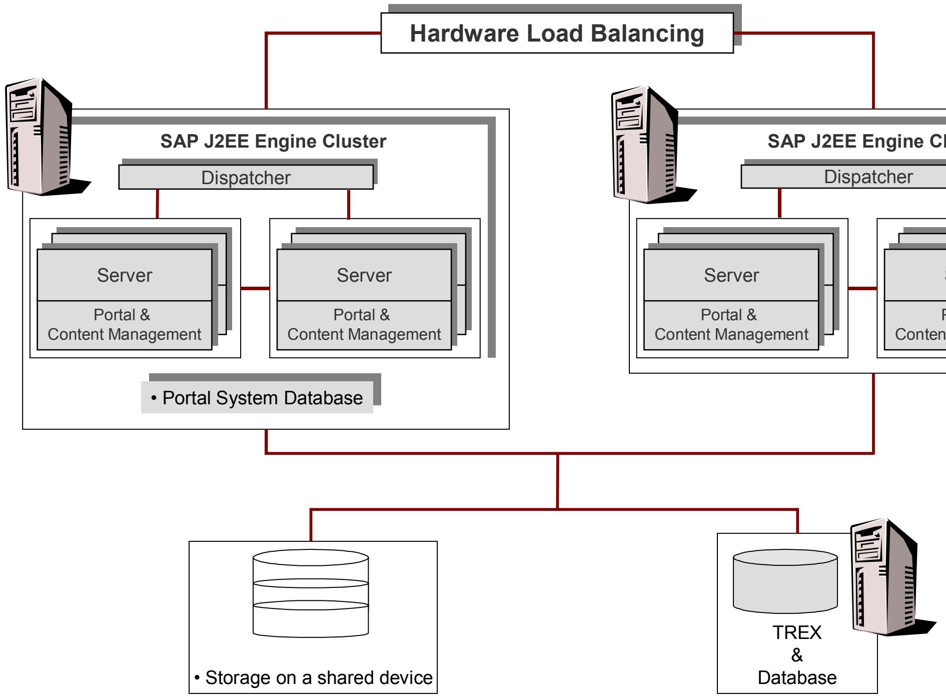


- Multiple dispatcher nodes, and several server nodes configuration

This configuration is most suitable for handling a large number of requests, over several machines. In this solution, you need to introduce a third party load-balancing solution in addition to SAP J2EE Engine.

A dispatcher node can be dedicated to a particular type of application. The optimal reduction of response time to client requests can be achieved because each dispatcher connects to a server in the cluster. In this way, the load factor on each server is easily balanced.

The following figure illustrates a cluster configuration in which there is more than one dispatcher node, and several server nodes with a third party hardware load balancing solution:



Note that connections within a local host are faster and efficient, complementing the performance of the network host-to-host connections.

In the above figure, the database is accessible to both SAP J2EE Engine installations. If one server fails, the clustering solution relocates the database package, including its shared resources, and automatically restarts it.

SAP J2EE engine is secured from failing by using a clustering solution that implements an IP address system that can be relocated. Such an IP address can be switched between servers. Requests from portal clients use this IP address to access the portal. If one server fails and shuts down, the portal environment can still run, as the entire deployment including the IP address is run by the other server.

Also, each server in the portal cluster needs to access shared data in the central storage, the global directory, which stores configuration information for portal. A highly available implementation can be achieved when the shared data is located in a central storage.

For additional information on the configuration settings for optimizing performance in the cluster environment of SAP J2EE Engine, refer to the guide, *How To Fine-Tune SAP Enterprise Portal 6.0* at: **service.sap.com/ep60,** *Enterprise Portal 6.0 → Documentation & More → How-To Guides.*

Performance and Availability in the Cluster

Using only SAP J2EE Engine, you can achieve highly available, reliable, and enhanced performance for SAP Enterprise Portal. In addition, you can implement other load balancing solutions for the portal in addition to the dynamic load-balancing feature in SAP J2EE Engine.

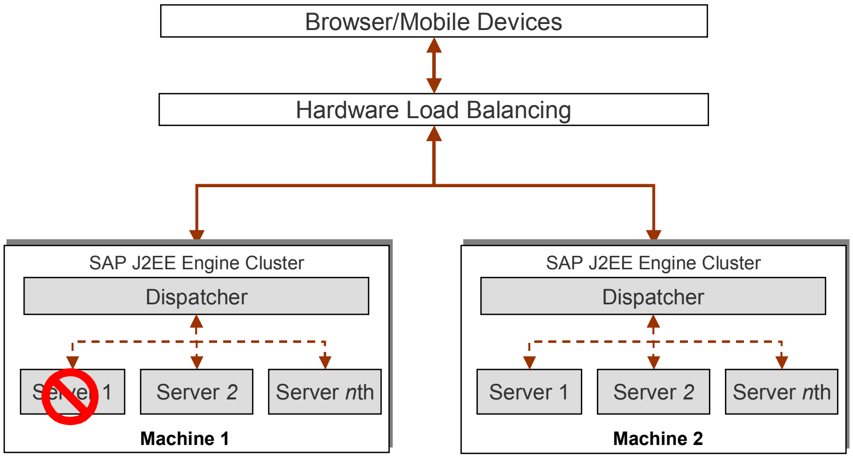
SAP J2EE Engine include the following features:

- **High Availability and Reliability**

  When a Server shuts down in the network, the Dispatcher automatically detects the failed server, and then loads the portal on another node, thereby minimizing system disruption. For more information, refer to Failover Recovery Mechanisms [Page 27].

  The following illustrates the failover infrastructure in the cluster configuration:

  SAP J2EE with a Third Party Load-Balancing Solution



- **Performance and Scalability**

  SAP J2EE Engine implements load balancing, and enables scaling across multiple servers. That is, you can add extra machines with additional portal installations to the existing cluster environment, as portal use increases.

  In addition, SAP J2EE Engine load-balancing can handle a large number of requests, and provides shorter response times.

  For more information, see Planning the Cluster Configuration [Page 21].

- **Manageability**

  SAP J2EE Engine enables the portal administrator to monitor and to manage cluster resources on all machines from a central location.

  It has a DBMS service that controls the SAP J2EE Engine deployment database. This database contains information about deployed Web applications on the SAP J2EE Engine server.

  To maintain the same applications across the SAP J2EE Engine nodes, this database is synchronized across all nodes. The deployment database can be synchronized by designating a SAP J2EE Engine Server node as a primary or a non-primary on a host.

  A primary DBMS service contains a local copy of the deployment database, which is synchronized against the other nodes. A dependent DBMS service does not contain a local copy of the deployment database, but reads the database remotely from a primary DBMS service.

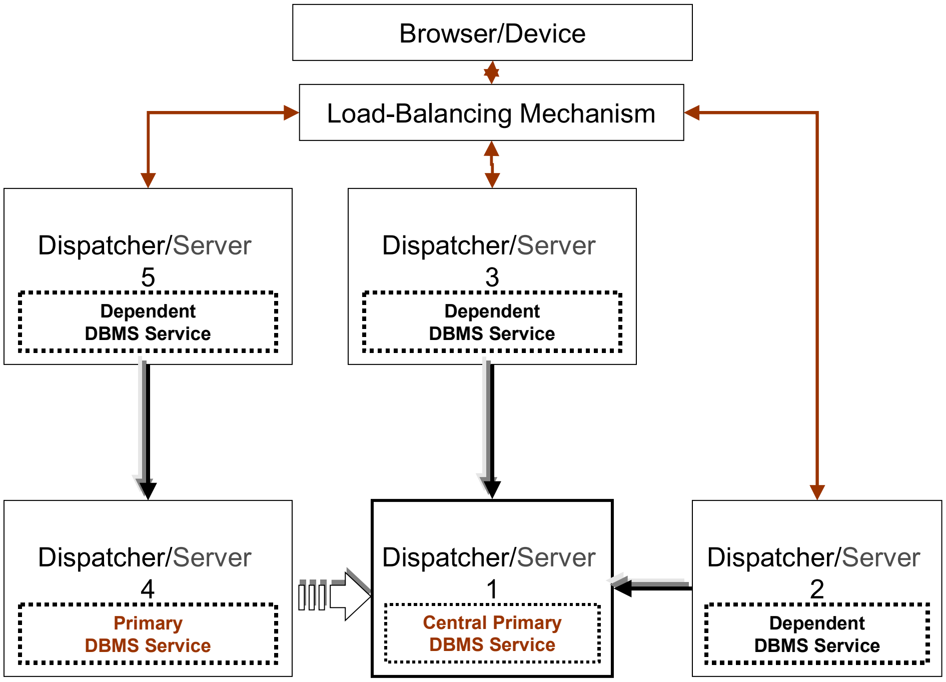
SAP J2EE Engine can have several primary servers, and several dependent servers. There is the need to have at least one primary Server node at all times, since the DBMS synchronization takes place at startup.

To optimize DBMS service operations, we recommend that you configure all other Server nodes in a cluster as dependent.

If you configure only one primary Server node, and it fails, any Server nodes dependent on it cannot restart until the primary server is back, before they can join the cluster again.

For this reason, you need to configure two or more primary Server nodes, so when a central primary Server node crashes, other primary Servers can take its role and synchronize the dependent nodes.

The following illustration shows an example of a central point of the cluster connectivity, with primary and non-primary Server nodes:



For additional information on how to configure and optimize the SAP J2EE Engine processes for SAP Enterprise Portal, refer to the guide, *How to Fine-Tune SAP Enterprise Portal 6.0*, at: **service.sap.com/ep60,** `Enterprise Portal 6.0 > Documentation & More > How-To guides`

SAP J2EE Load-Balancing Mechanism

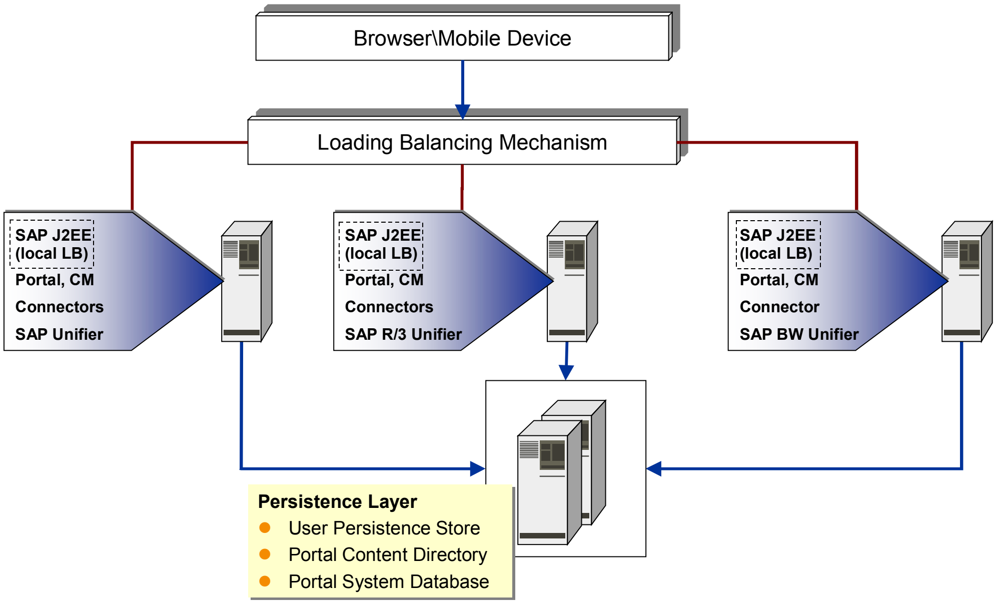
Load balancing is dividing the work of one server machine among several servers, so that more work gets done in the same amount of time and all clients get served faster. In addition to the SAP J2EE Engine load balancing, you can implement other hardware or software load balancing solutions.

With SAP J2EE Engine, you apply load-balancing to the portals distributed across several machines, thereby dividing their workload.

The load balancing system of SAP J2EE Engine implements client requests in relation to the Server nodes in the cluster. By doing so, the full capacity of each server can be reached. The system automatically detects the least occupied server, and redirects the request to that server for further processing.

The following is an illustration of the portal cluster in a SAP J2EE load-balancing configuration:

Distributed Portals Accessing a Single Persistence Layer



Failover Recovery Mechanisms

The applications you deploy on one machine are automatically mirrored onto other servers within the same SAP J2EE Engine environment. Therefore, if a server fails, SAP J2EE can redirect client requests to other servers for the same application, providing users with continuous service.

There following are some of the failover recovery features in SAP J2EE Engine:

- Simple Failover Recovery (SFR)

  Simple failover recovery is performed when a server fails. In such cases, the inherent administration mechanism reroutes the client requests to another Server node.

- Session-Level Failover Recovery (SLFR)

  Session-level failover recovery is based on the persistent engine of SAP J2EE Engine.

  This failover recovery is achieved using synchronous replication, that is, session data is replicated on specified Server nodes simultaneously. If the server processing the client request fails, the persistent engine reroutes the request to another server on which the services session data has been replicated. In this way failover recovery is achieved, and Server node overloading is prevented.

Preparing SAP Enterprise Portal Cluster

There are several configurations for the SAP Enterprise Portal cluster environment. However, you must prepare each environment with the prerequisites and requirements of the portal.

For detailed information on the prerequisites and requirements of the portal, see the *SAP Enterprise Portal Installation Guide*, at: **`service.sap.com/ep60,`**
navigate to *Enterprise Portal 6.0* > *Documentation & more* > *Installation*.

To prepare the cluster environment for the portal, do the following:

1. Perform all prerequisites for SAPinst and the portal on the portal host. You can dedicate separate machines to both the user persistence store and the portal database.
2. Deploy the SAP Enterprise Portal on a machine other than the machines dedicated to the portal database or the user persistence store. This portal becomes the development and testing environment.
3. Perform the relevant post installations and configurations for the development and testing environment.

   For the portal cluster, you need to share the global portal folders in the local file system.
4. Plan and configure each additional machine for deployment of SAP J2EE Engine.

   For detailed information, refer to the *Additional Clustered Portal Installation Guide* at: **`service.sap.com/ep60,`** and navigate to *Enterprise Portal 6.0* > *Documentation & More* > *Installation*.
5. Deploy additional instances of SAP J2EE Engine on each machine.

   After, configure SAP J2EE Engine from any of the machines in the cluster.
6. On each machine, mount the shared global portal folders using the same path.
7. Reset the SAP J2EE Engine.

See also:

Planning SAP Enterprise Portal Cluster [Page 28]

Clustering SAP Enterprise Portal

When demand on the existing portal increases, you can add extra machines and quickly scale the portal to meet the new demand.

You configure extra machines for deploying the portal by installing an additional instance of SAP J2EE Engine on each machine.

For detailed information on installation of additional instances of SAP J2EE Engine, see the *Additional Clustered Portal Installation Guide* at: `service.sap.com,` and navigate to *Enterprise Portal 6.0* > *Documentation & More* > *Installation*.

Recommended Cluster Setup For Enterprise Portal

The following is the recommended technical infrastructure on how to setup the SAP J2EE Engine for the Enterprise Portal:

- Setup one primary Server node (**DependentElement = false**) to serve as a central point of connectivity for the cluster environment. All other Server nodes in each SAP J2EE Engine host must be non-primary (**DependentElement = true**).
- In a productive environment you can set up additional primary Server nodes in other SAP J2EE Engine hosts as a backup to the central primary Server node.

For additional information on the settings for SAP J2EE Engine configuration for the portal, refer to the guide, *How To Tune Enterprise Portal 6.0* at: **service.sap.com,** navigate to *Enterprise Portal 6.0* > *Documentation & More* > *How-To Guides*

- Never change the cluster ID of a Server node.

  If you do, you **must** redeploy all applications deployed on this node.
- Never leave the cluster with non-primary (**DependentElement=true**) servers only.

  If this happens, you must restart the cluster immediately.

  If you cannot restart the cluster, you will need the help of Enterprise Portal technical support.
- Never change manually the data files of the DBMS service.

Identifying Single Points of Failure (SPOF)

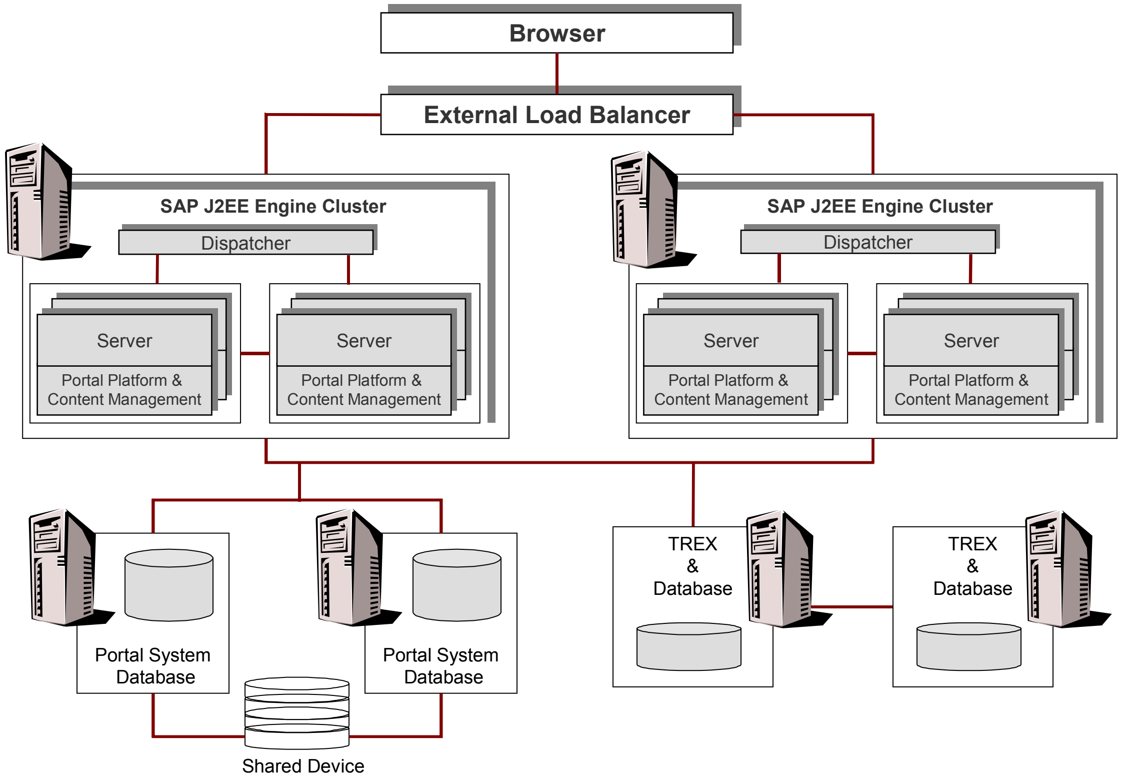
By identifying and eliminating all single points of failure (SPOF) in the portal landscape, you ensure a continuous and highly available portal environment.

Single points of failure occur where only one software component is deployed for specific tasks in the portal landscape. For example, a single portal system database represents a single point of failure. If that single component fails, there is no alternate one to take its place.

The following represent identified single points of failure in the portal landscape:

- Portal System Database
- SAP J2EE Engine and portal installation folder, */usr/sap/<sid>/global*
- Running a single SAP J2EE Engine dispatcher node, in the absence of a balancing mechanism
- Deployed load balancing mechanism in the cluster

The following illustration shows a high availability portal landscape:



In the above illustration, the portal system database, TREX and its databases, and SAP J2EE Engine are clustered with a load balancing mechanism, and the global directory in the file system is in the shared device.

Portal System Database

The portal system database is comprised of database schemas for the Portal Content Directory (PCD), User Management (UM) and Content Management (CM). Some of the schemas contain elements used at runtime in various processes of the portal.

If one portal system database is deployed and used by all the portals in a cluster, it represents a single point of failure. When the database server fails, it shuts down the portal.

Using the specific vendor's recommendation for clustering the database, or other third party solutions, can ensure high availability of the of the portal.

Portal Installation Folder

In a cluster environment, all the portals access central configuration data on a single physical server, in the folder, */usr/sap/<sid>/global*.

This is the global directory which stores the configuration information for the PCD and CM. Each portal in the cluster needs to access this configuration information.

To remove this SPOF, use an external disk together with a clustering solution.

Single SAP J2EE Engine Dispatcher Node

The portal platform runs on top of the SAP J2EE Engine. If access to the portal platform is through a primary dispatcher node of the SAP J2EE Engine, this can fail and cause the portal to shut down.

To remove the SPOF represented by the single dispatcher node, you can use multiple dispatcher nodes with a third party load balancing solution. However, this solution only shifts the SPOF to the load balancing solution.

Load Balancing Mechanism

A load balancing solution allows access to the portal environment, such that all incoming requests are distributed to the different SAP J2EE dispatcher nodes according to definitions of the implemented solution.

Making the load balancing mechanism highly available is recommended.

Knowledge Management in the Portal Landscape

When you run the SAPInst installation tool, you can optionally choose to install Knowledge Management. This documentation gives you detailed information on how you can integrate Knowledge Management (KM) in the portal landscape. The information is particularly useful, if you have already installed the portal and want to install KM at a later point in time. It focuses on integrating KM in system landscape that is set up with or without load balancing.

KM is comprised of 2 main components:

- Content Management (CM)

  CM enables resources that are stored in heterogeneous repositories to be managed in a uniform manner with the help of a variety of user-centric services.

  For more information see Content Management [Page 33]

- Retrieval and Classification (TREX)

  TREX enables the search and classification operations that are a core part of the CM application.
  For more information see TREX [Page 40]

If you decide to install KM, you have to install both the Content Management and TREX components.

Content Management

When you plan the portal landscape, you can distribute the CM and TREX components and their subcomponents to machines in various ways. To understand the distribution options, it is useful to first identify the main parts of the CM application.

CM is comprised of:

- The Content Management application

  The application consists of services that provide the runtime environment for the execution of KM functions.

- The database running on a database server

  The database stores the CM data. This includes the contents of the CM repository, access control lists for resources and repository metadata.

- The configuration data

  The data consists of parameters that control the availability of services and the behavior of CM. They are stored on the file system in a shared directory.

  After the installation of CM, various types of repositories can be configured. The contents of these can be stored at different locations, for example, on a file system, on a web server or on a Lotus Notes server. The repository data of a productive CM system is therefore generally distributed over several machines.

Content Management Distribution

Content Management (CM) is closely integrated with the portal. As a consequence, the location of CM in the portal landscape is generally dictated by the location of the portal. The CM application must be installed on the same machine as the portal. The shared configuration data must be stored at the location prescribed by the portal and the CM database is usually installed on the machine where the portal database server is located. In systems with a high workload, the CM database can alternatively also be installed on a separate machine with a database server that is set up exclusively for CM.

In contrast to the above, TREX is an autonomous component that does not have to take the location of the portal into account and is usually set up on one or more separate machines. The only requirement is that a connection is established to CM and the portal during or after the installation.

The optimal location for CM and TREX in the portal landscape depends on the size of the system. The following gives you recommendations for small, medium-sized and large systems. It focuses on:

- Test system
- Production system without load balancing
- Production system with load balancing

⚠

Keep in mind that the recommendations here are only general guidelines. An optimal distribution of components is normally worked out in a sizing procedure that carefully analyses the expected workload and takes into account the processing power of available machines.

Test System

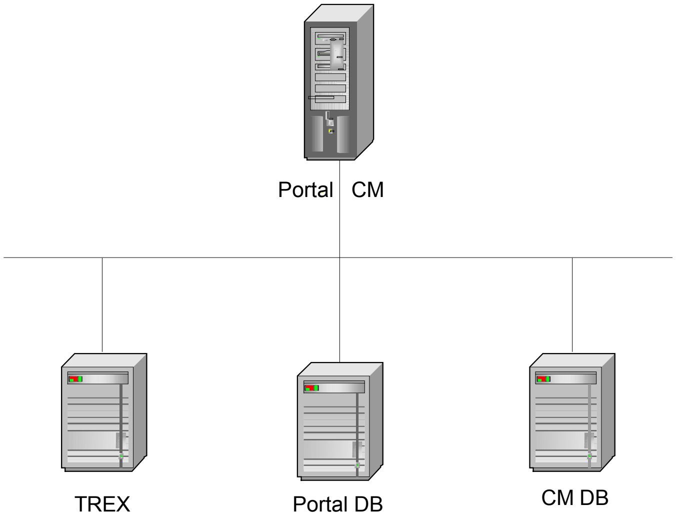
In a small system that is only used for test or demo purposes, the portal and KM components can run on a single machine. The portal, CM application, TREX, and database server can all be installed on the same machine. No distribution of components is necessary.

Production System

In a standard hardware configuration, for a medium-sized production system without load balancing, the CM application and database are normally located on different machines. The CM application is installed on the same machine as the portal. The CM database is installed on the machine where the portal database server is located or on another machine where an additional database server for CM is installed.

In medium-sized production system, TREX is normally installed on a dedicated machine that is used exclusively for TREX. In some cases, it can be installed on the same machine as the portal and CM. However, installation on a single machine with CM and the portal is only permitted if sufficient memory is available for all the components. A maximum of 6GB memory may be required exclusively for TREX. If insufficient memory is available, this can severely degrade performance.

The graphic shows a system that uses a dedicated machine for the portal and CM databases.



Load-Balanced Production System

A portal that has to serve a large number of clients can be set up in a cluster that uses load balancing to distribute client traffic among machines. If the portal is set up in a cluster, CM must be integrated in the cluster solution. An instance of the CM application **must** be installed on each of the machines where a portal is installed. The number of CM and portal instances must be the same.

In a cluster, the load balancer spreads the workload across all the available CM instances which simultaneously process the requests of clients. As a result, CM is able to handle large workloads, avoid performance bottlenecks and grow incrementally with the system load. When the demands of clients increase, new machines with CM instances can be added to increase throughput and performance.

If a CM machine fails, the other CM instances can still continue processing. The load balancer identifies the failure and automatically redirects requests to the remaining machines. However, the load balancer does not support failover. If a CM fails on one machine, the processing of client requests is aborted and not automatically transferred to another machine. Also, if the tasks of a service have been scheduled to run on a specific CM instance that fails, they must be rescheduled to run on another instance before processing can continue normally. The cluster can heighten the availability of the system, but not guarantee uninterrupted continuous processing of all requests.

In summary, load balancing in a cluster has the following main advantages. It enables CM to:

- Handle large workloads that cannot be processed effectively by a single machine
- Increase processing capacity incrementally as the workload grows
- Improve availability
- Flexibly change the system landscape to meet requirements

Content Management in a Cluster

This section examines how Content Management (CM) components can be integrated in a portal landscape that is set up as a cluster. It also explains load-balancing mechanisms and the strategies that CM uses to enable several instances to simultaneously process and access the same data.

The main components of CM in a cluster are:

- Several instances of the CM application
- A CM database
- Shared directory on the file system

As mentioned earlier, an instance of the CM application **must** be installed on every machine in the cluster where a portal is installed. Each instance accesses **the same set of persistent data** that is stored in the database and on the file system. The database stores the CM resources, their metadata and access control lists. The directory on the file system stores the configuration data.

The CM load in a cluster is spread across the available machines with the help of two components:

- An external load balancer or, alternatively, the dispatcher of the SAP J2EE engine
- A scheduler service
-

Load Balancer

The load balancer receives all the requests coming from clients and distributes them to the CM instances. Depending on the type of load balancer, requests are distributed in a round-robin procedure or on the basis of the CPU load. If distribution is based on the CPU load, a new request is always directed to the machine with the lowest CPU load. In this way the workload is evenly shared amongst all the available machines.

Scheduler Service

The CM scheduler service also distributes the load in a cluster. However, it operates independently of the load balancer. Whereas the load balancer handles requests coming from clients, the scheduler handles tasks that originate from internal CM services like subscription, indexing and content exchange. It directs the tasks of services to be executed on a specific CM instance. As these tasks can generate a high system load they should also be optimally distributed.

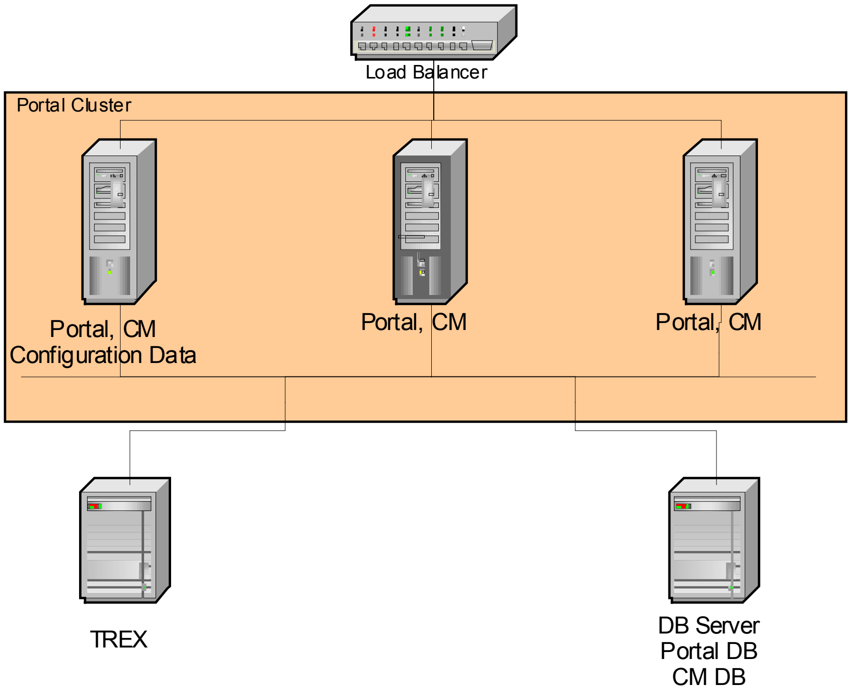
The scheduler itself does not decide where service tasks are run. After a CM installation, the administrator has to use the settings offered by the scheduler to assign tasks to a specific CM system. It is then the responsibility of the scheduler to allocate the tasks to the correct system at runtime.

Integration of CM in a Cluster

The following graphics illustrate two different ways in which CM can be integrated in a load-balanced portal landscape.

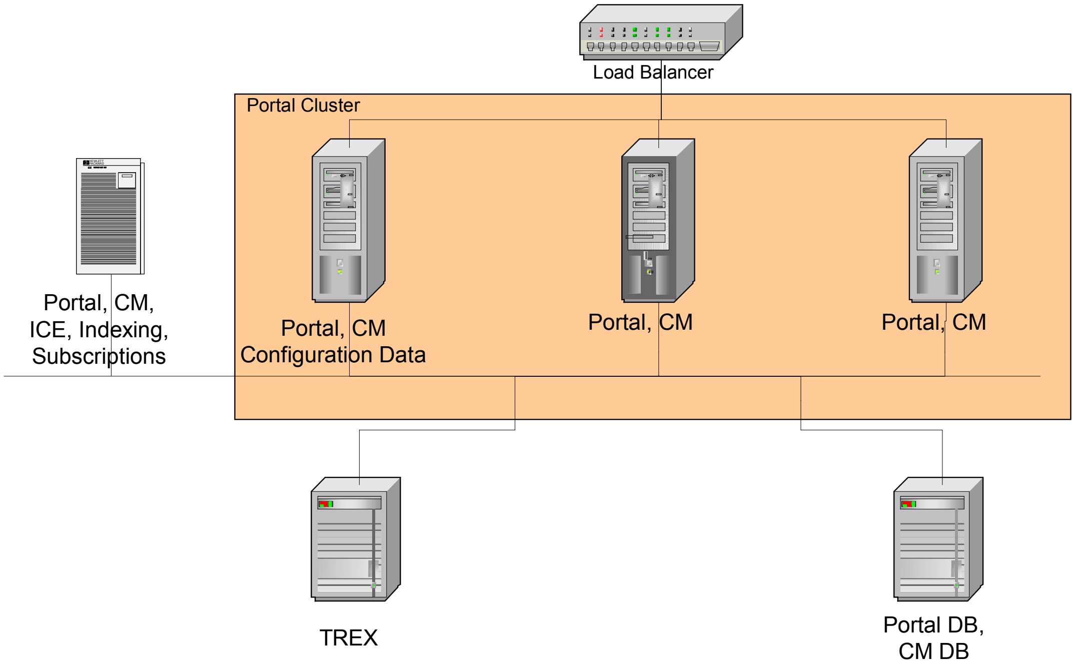
The first graphic shows a standard installation where the location of CM is dictated by the location of the portal. An instance of CM runs on each of the nodes where a portal is installed. The configuration data is stored on the first cluster node. The database server, with the CM and portal

schemas, is located on a machine that is excluded from the load-balancing logic.
All the CM instance access the same configuration data and database server.



The second graphic shows an alternative setup that includes an additional machine that is excluded from the load-balancing logic. This machine runs a portal and CM instance, but is not used for processing client requests. It is reserved for processing the tasks of CM services that generate a high system load like content exchange, index management and subscriptions. The CM scheduler service ensures that the tasks of the services are directed to the additional machine.

Whether it is advisable to include an extra machine that exclusively handles the tasks of CM services depends on the individual requirements of a system. If, for example, a very high workload is generated by the CM index management service, then the integration of an additional machine would improve performance because it would leave the other machines free to process client requests.



In the graphics above, the CM and portal database are located on the same machine. Optionally, the CM database can be installed on a separate, dedicated machine.

Data Synchronization Strategies

As several instances of the CM application access and process the same data, the CM load balancing solution must ensure that each instance uses current data. If one instance changes data then synchronization mechanisms must ensure that other instances that have cached the data are aware of the changes. Different mechanisms are used for synchronization, depending on whether the data is stored on the file system or on the database.

Synchronization of Configuration Data

The configuration data for all the CM instances is stored on a file share. The share contains data that is used by all instances and data that is unique to individual instances.

The data that is unique to individual instances is stored in a property file and determines how that specific instance is configured. As the property file is unique for each instance, there are no conflicts. The instances can easily identify, read and update their own data without encountering any conflicts with other instances.

When the configuration data is shared by all instances, the situation is more complex. In this case, when data is changed, the configuration framework triggers events to inform all instances of the change. Each application running on an instance is responsible for receiving and reacting to such events. If an application registers an event signaling that data has changed then it must read the new data from the file share. This strategy ensures that the application always uses the most current data, even if multiple instances are changing the same data.

Synchronization of Caches

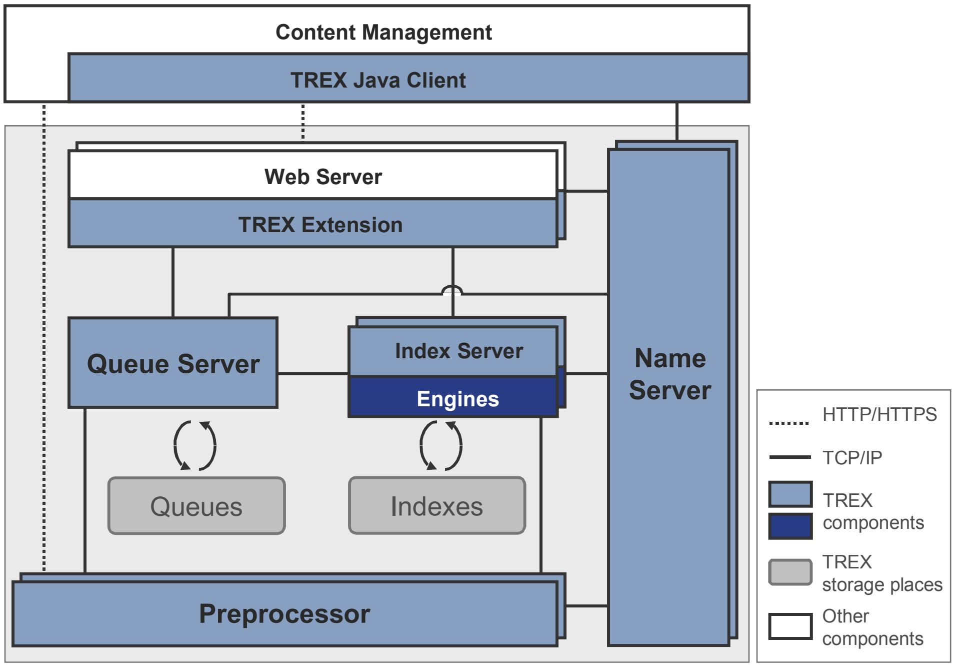
To speed up processing, each KM instance caches data from the database in a local memory cache. To ensure that each cache works with the same data, a synchronization strategy that uses timestamps has been implemented.

TREX

TREX processes all index, search, and text-mining requests received from the portal. The following are the TREX components:

- Java client
- Web server with TREX extension
- Queue server
- Preprocessor
- Index server
- Name server

The graphic below shows the individual components and how they communicate.



Java Client

TREX provides several interfaces that can be used to integrate TREX functions into an application. The Java client is an interface for Java applications.

The Java client is integrated into Content Management. This means that the TREX functions are available in Content Management and in the portal.

Web Server with TREX Extension

Content Management (more precisely, the Java client) accesses the TREX functions using a Web server. Communication between Content Management and the Web server takes place using HTTP/HTTPS and XML. The Web server receives requests and forwards them to the index server and queue server. The servers then process the requests.

A TREX component that extends the Web server with TREX-specific functions is installed on the web server. Technically speaking, this component is realized as follows:

- On Windows, as an ISAPI server extension for the Microsoft Internet Information Server
- On UNIX, as a shared library for the Apache Web server.

Queue Server

The queue server enables the asynchronous indexing of documents. It has a separate queue for each index. It gathers documents to be indexed into the queues. It transfers documents to the index server for the actual indexing process at regular intervals.

You can use the queue parameters to control when and how many documents are transmitted. This allows you to schedule indexing for times at which the index server does not receive a large amount of search requests.

The queue server forwards the documents to the preprocessor before transmitting them to the index server.

Preprocessor

The preprocessor has two tasks:

- When a search takes place, the preprocessor carries out a linguistic analysis of search queries. The preprocessor passes the results of the analysis to the index server, which then processes the query further.
- When indexing takes place, the preprocessor prepares the documents for the indexing process. The preparation consists of the following steps:
    - Loading the document

      In the portal, documents are not normally transferred directly to TREX. Instead, they are forwarded in the form of a URI that references the storage location of the document in question. The preprocessor resolves the URI and collects the actual document from the repository.
    - Filtering the document

      Documents can exist in various formats (Microsoft Word, Microsoft PowerPoint, PDF, and so on). The preprocessor filters the documents, that is, it extracts the text content and converts in to Unicode format UTF 8 for further processing.
    - Analyzing the document linguistically

      The preprocessor uses a lexicon that analyzes texts in various languages.

Index Server

The index server is responsible for indexing, classifying, and searching. It receives requests and forwards them to the TREX engines. The engines provide the actual core functions of TREX:

- The search engine is responsible for standard search functions such as exact, error-tolerant, linguistic, Boolean, and phrase search.
- The Text-mining engine is responsible for classification, searching for similar documents (‘See Also’), the extraction of key words, and so on.
- The attribute engine is responsible for searching in documents properties such as author, creation date, change date, and so on.

Name server

The name server is used with large distributed TREX installations. It uses its databases to store and coordinate system-wide information. It also ensures that the TREX servers can communicate with each other and that TREX can communicate with Content Management. The name server is also responsible for distributing the system load if more than one TREX server is capable of carrying out a task.

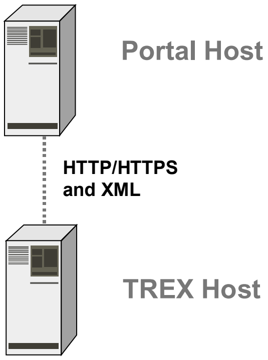
In a distributed scenario, you can install several name servers to ensure that a name server is always available. A replication procedure ensures that the databases of the different name servers are synchronized.

TREX Distribution

TREX offers a flexible architecture and can be adapted to different requirements. You can scale TREX if necessary. Your options range from a minimal system with one host, to a large distributed server landscape.

Single-Host System

A minimal TREX system consists of a single host that provides all TREX functions (indexing, classification, and searching). You can use a minimal system as a demo and test system, or as a production system. For a production system, we recommend that you install TREX on a dedicated host that is used exclusively for TREX.



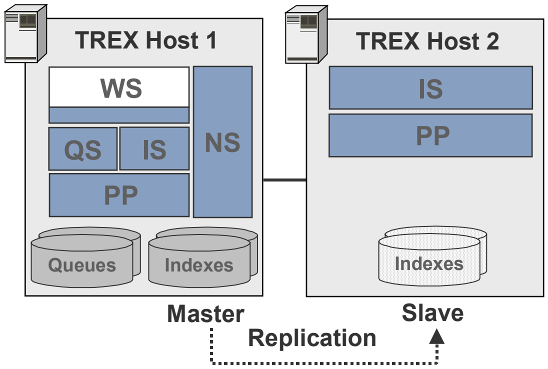
Multiple-Host System

In most cases you first set up one TREX host. You then have numerous options for scaling TREX. You use a scaled scenario to distribute the search and indexing load among several hosts and to ensure the availability of TREX.

In a multiple-host system, the individual hosts are responsible for different tasks depending on which TREX components run on them. For example, you can set up dedicated search servers with copies of the original indexes and configure automatic index replication to keep the copies up-to-date.

Example

The graphic below shows a distributed TREX system with two hosts:



IS = index server, NS = name server, PP = preprocessor, QS = queue server, WS = Web server

The first host is responsible for indexing, classification, and searching. The second host is exclusively a search server, and is not used for indexing or classification. This host only stores copies of the original indexes. These copies are updated using index replication.

For details on distribution options and implementation, see the Guide *Scaling Retrieval and Classification (TREX)*. It is part of the zipped package, *EP6.0 SP2 Installation Guide (Portal, CM, & Collab),* at: **service.sap.com,** and then navigate to *Enterprise Portal 6.0 > Documentation & More > Installation*

Security

Access to most sources of information requires end users to be authenticated. Since information sources may contain sensitive data, it is important that users can be identified and their identity authenticated.

Generally, users access the portal with security credentials configured for the local computer. Using portal tools, you can configure user-related data for logging onto the portal. When successfully authenticated, users can access applications, information and services in the Enterprise Portal to which they have permissions using Single Sign-On.

Licensing the Clustered Enterprise Portal

A temporary license is automatically installed upon installation of the initial portal platform. However, you can obtain a permanent license that should be installed soon after the portal platform installation completes.

The license is installed in the portal database, which is common to all the machines in the cluster, therefore:

- All the machines in a portal cluster are covered by a license installed on any of them.
- When adding clustered portals, no changes are required concerning the license.
- Only if you want to create failover scenarios do you have to install additional permanent licenses.
- The license is cached for one hour, so it is possible to stop and restart the portal on which the license was installed. It should not be stopped for longer periods of time.

  For high availability of the portal, SAP recommends that you install a license on several portal machines.

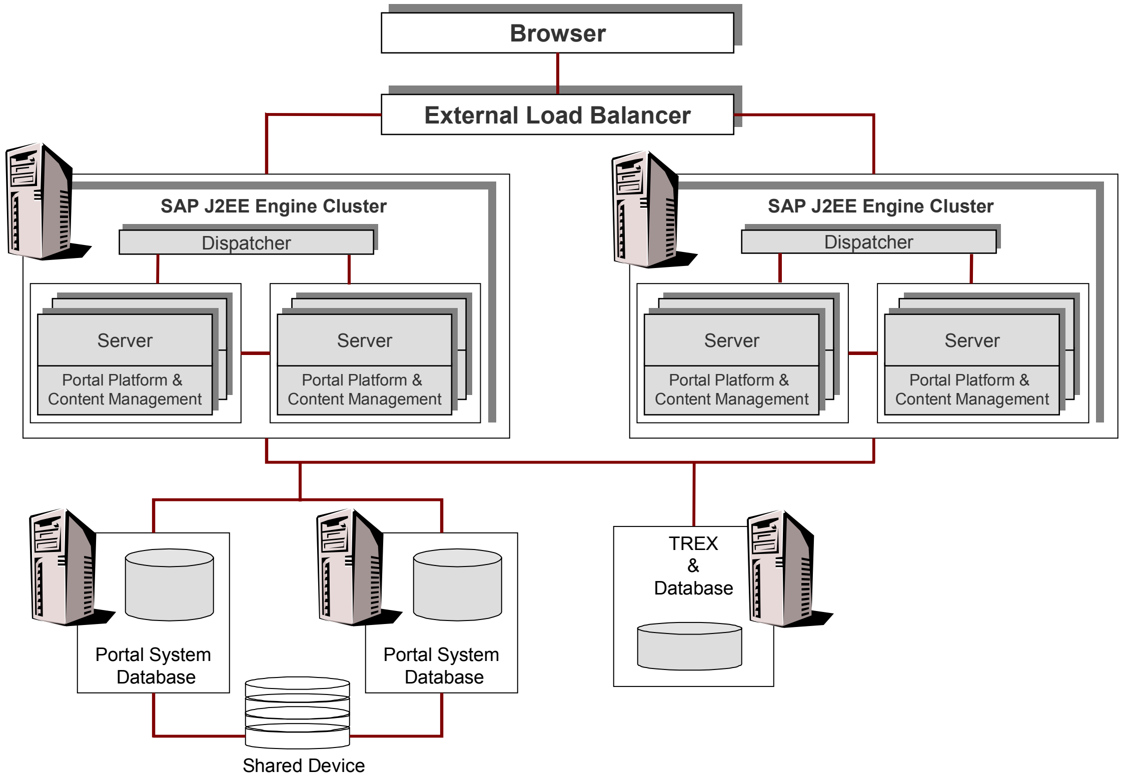
For detailed information on portal licensing, go to *EP6.0 SP2 Installation Guide (Portal, CM & Collab),* at: **service.sap.com,** and navigate to *Enterprise Portal 6.0 > Documentation & More > Installation*

Network Landscapes

When building the appropriate portal landscape, keep in mind the following portal requirements:

- Portal and Content Management must reside on the same machine.
- The different elements of the persistence layer, such as Portal System Database, and the User Persistence Store can be distributed among several machines.
- Firewalls must allow access according to the network connections and topography.
- The portal must have a single HTTP/HTTPS entry point.
- The two components of Knowledge Management, Content Management (CM) and TREX, must be integrated in the portal landscape keeping the following restrictions in mind:
    - CM is closely integrated with the portal. For this reason, the location of CM in the portal landscape is dictated by the location of the portal middleware and persistence layers. The CM application must be located on the same machine as the portal server. The CM database schema must be located on the same machine as the portal database.
    - For production systems, it is strongly recommended that TREX be installed on one or more dedicated machines that are only used for TREX.
      In exceptional cases, if enough memory is available for all portal components, TREX can be installed on the same machine as the portal server and CM. In this configuration, a maximum of 6GB of memory must be exclusively available for TREX.

The following drawing shows a typical portal landscape in the network environment:

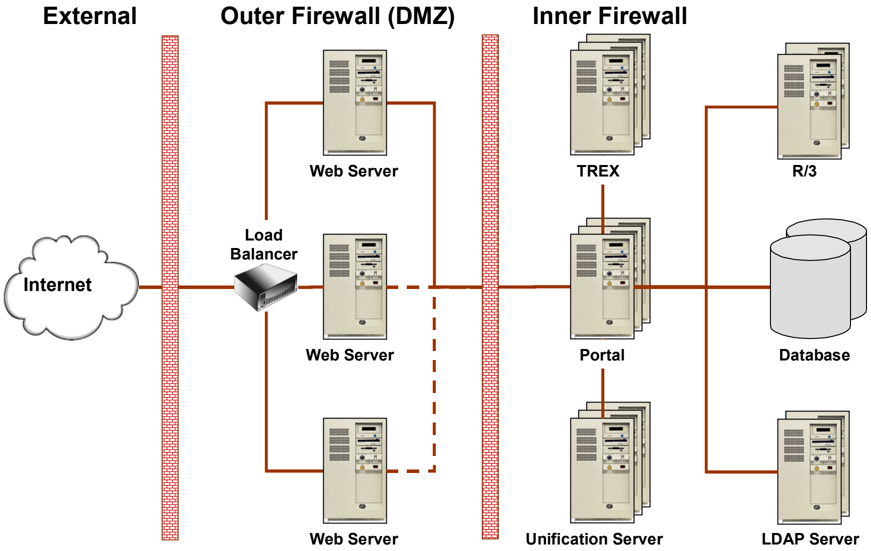


Distributed SAP Enterprise Portal can run behind a firewall to provide increased security while processing client requests. In this case, you deploy a reverse proxy server that obtains requests from portal clients and forwards them to the portal. Responses from the portal also go through the reverse proxy server, which delivers them to the clients.

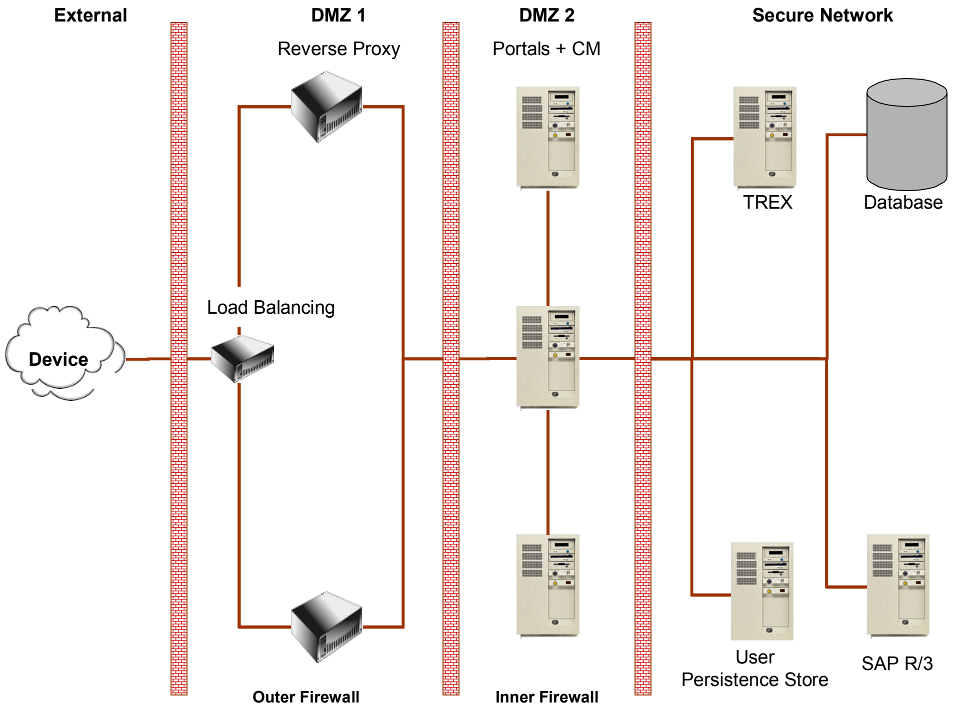
For additional information on configuring https (SSL) access to the portal using a reverse proxy, see **SAP Note 480 520.** Attached to this note is a document on how to configure Apache Web server with SSL access to the portal.

Detailed information on how to use the reverse proxy filter mechanism installed with SAP Enterprise Portal is located at: **service.sap.com/ep60**, *mySAP Enterprise Portal → Enterprise Portal → Documentation & More → How-To Guides → How to Configure the Reverse Proxy Filter for SAP Enterprise Portal 6.0 SP2*.

The following figure illustrates the production environment of SAP Enterprise Portal in a firewall:



The following figure illustrates the distributed deployment of SAP Enterprise Portal in a firewall:



The administrator must configure the firewall in accordance with the necessary free ports for the corresponding services, such as HTTP(s), P4, and so on.

Portal and Unification Servers in the DMZ

Purpose

In this landscape the middleware reside in the DMZ and receive all the HTTP/HTTPS requests from the Internet clients. They then process the requests and return the results to the clients. There is a direct connection between the portal Web server and the clients through a single HTTP/HTTPS port in the firewall.

- Unification Server must have a single HTTP entry point for each project. This entry point is the IIS port on which the project was created. Therefore, a Unification Server machine requires one port per project to be opened through the firewall.
    - If you have several different projects, then open a port for each project in the outer firewall, between the client and the reverse proxy.
    - If you have several different projects in the DMZ and you do not want to open several ports, do the following:
        - Locally enable each unifier project to use the same port as the target portal in which you intend to run the projects.

    ⚠

    Installing Unification Server on the same machine as the portal is not supported for production systems. We recommend using a separate machine for Unification Server.

    The persistence layer of the portal and Unification Server can, however, be installed on the same machine.

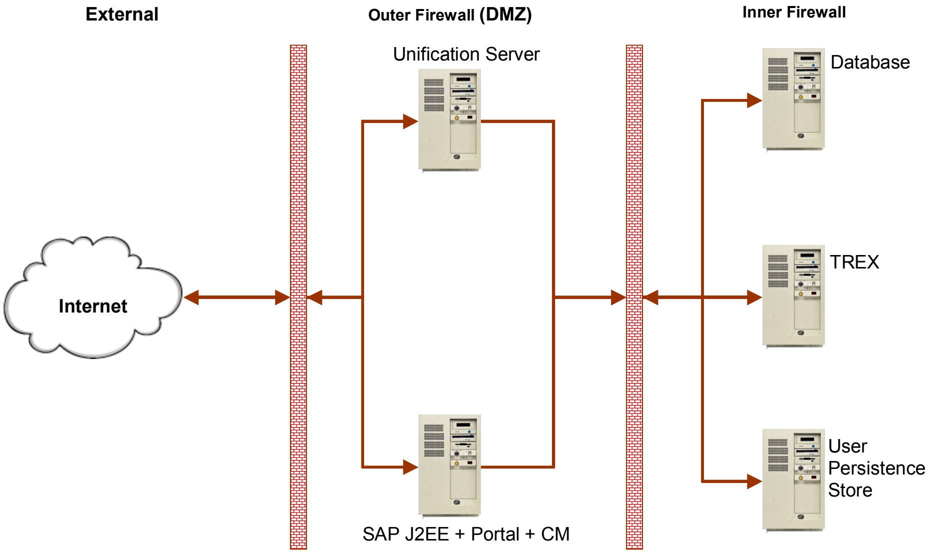
The persistence layer resides in the secured network. It is possible to move parts of the persistence layer components to the DMZ according to the security and backup policy of the customer.

In this landscape, the inner firewall separates the middleware and persistence layer. The following are the requirements of the inner firewall settings:

- LDAP directory port.
- If there are connections to back-end applications, the portal needs to access them through the firewall.

  If these applications have a Web interface in the DMZ, such as the Internet Transaction Server (ITS), the connection is to this interface.

The following illustration shows the SAP Enterprise Portal in a firewall configuration:



If a port cannot be opened for a resource in the inner firewall, that resource can be moved to the DMZ as well.

System and Portal Administration

In any infrastructure, there is the need to maintain the interoperability of the various systems, such as security administration, systems administration, and application administration.

Some portal operations such as URL redirection, firewall configuration, Domain Naming System (DNS), reverse proxy settings, and specifically, the configuration of the servers running the load balancing system all influence the infrastructure. Hence, troubleshooting requirements and responsibilities among the various systems such as network, system, and application administration must be clearly defined

SAP Enterprise Portal extends the security mechanism implemented by SAP J2EE Engine. Roles form the main mechanism by which users can access content in the portal.

Access to most sources of information requires that users are authenticated. Since information sources may contain sensitive data, it is important that users are identified and their identity authenticated.