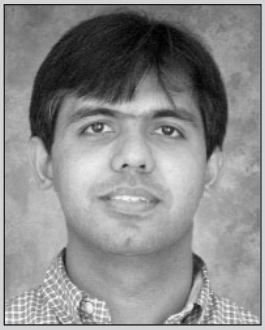
Designing a Solid, Lasting Landscape for Your SAP Enterprise Portal Implementation: Using the Most Effective Technical Options to Meet Your Key Requirements

Rizwan Uqaili



Rizwan Uqaili is an Engagement Manager with Business Intelligence Services at Rapidigm. He also manages Enterprise Portal Services at Rapidigm, and has assisted numerous clients with their strategic SAP Business Intelligence and SAP Enterprise Portal initiatives and implementations. Rizwan has been a key speaker at ASUG conferences as well as the SAP BW and Portals conference.

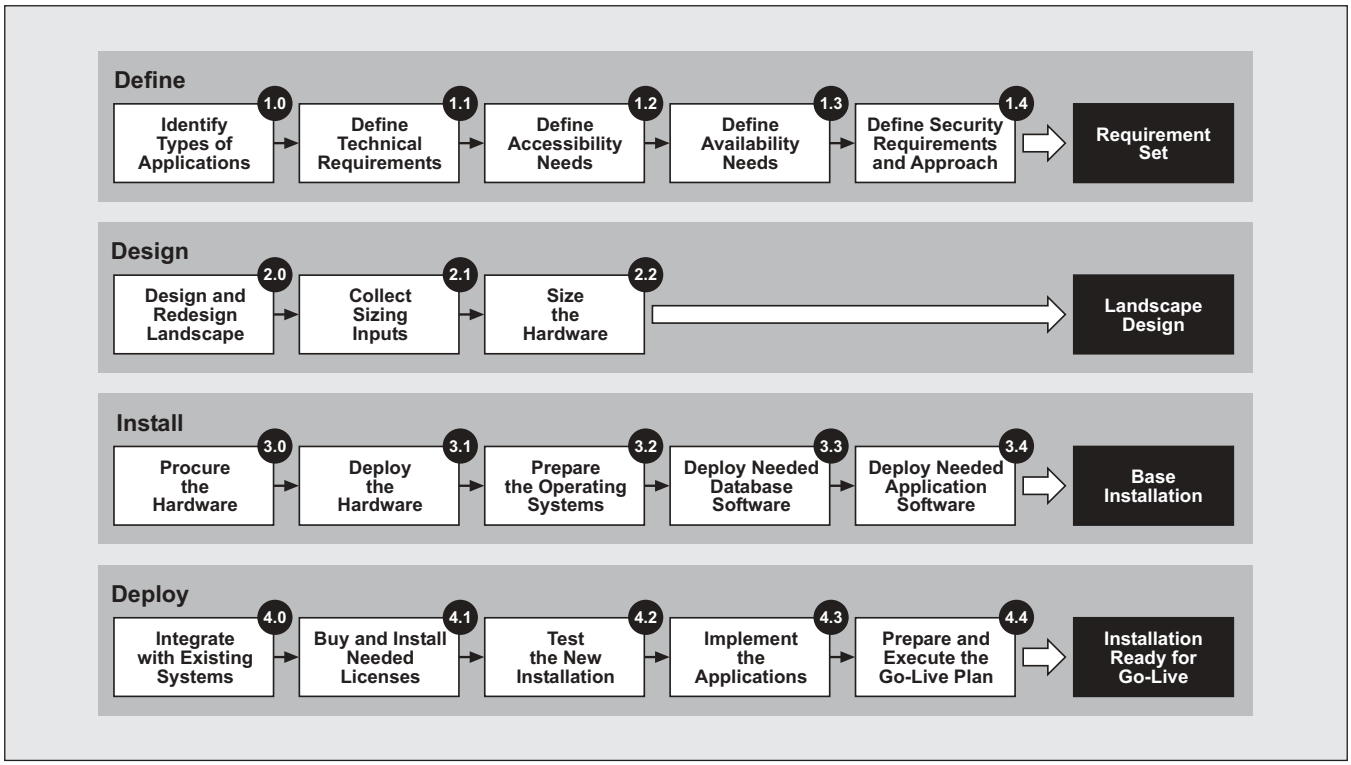
(complete bio appears on page 30)

SAP Enterprise Portal (SAP EP), now in version 6.0, offers users a single, convenient, personalized access point to enterprise applications and documents, and offers IT teams a centralized, Web-based infrastructure for handling complex issues like scalability, redundancy, security, user authentication, external access, and content administration. However, the pressure to get a portal up and running as soon as possible to take advantage of these capabilities can yield a landscape your company will likely outgrow in as little as 6 to 12 months. So how do you design a strategic, cost-effective portal landscape that meets both today's needs *and* tomorrow's needs?

The only way to keep pace with unavoidable, exponential portal growth without significant cost, rework, and production interruptions is to base your landscape design on a deliberate growth strategy. In the first installment of this two-part article series, I showed you how to precisely define your portal landscape requirements, and how to use these identified requirements to determine the most strategic and cost-effective initial portal landscape that will meet your needs.[1] I also explained the importance of designing your pilot portal landscape around longer-term objectives like providing Internet accessibility, offering GUI options like SAPGUI for HTML or SAPGUI for Windows, or shifting from service-level-based availability to 24/7 availability, even if your initial plans only include a single division running a single application. Chances are, at some point users will want more functionality, and other divisions will want in on the benefits of the portal, too — and probably sooner rather than later.

[1] "Designing a Solid, Lasting Landscape for Your SAP Enterprise Portal Implementation: Identifying Your Key Requirements and Understanding Your Design Options" (*SAP Professional Journal*, September/October 2004).

Figure 1 Implementation Methodology for SAP Enterprise Portal Projects



This second installment discusses the specific technical options that are available to you for designing and sizing your production portal landscape to meet your identified requirement set (I will also touch on some special considerations involved in designing development and quality assurance portal landscapes). Steps 2.0-2.2 in **Figure 1** summarize the design process (steps 1.0-1.4 were covered in the first installment).[2] The technical options available for addressing steps 2.0-2.2 are outlined in **Figure 2**, where they have been grouped into the six major design choices that will ultimately determine your landscape design — server position, deployment configuration, authentication method, authentication store, communication security method, and single sign-on strategy. I will discuss several of these choices in the context of five sample landscapes that accommodate common requirements, which you can use as the basis for your own portal design:

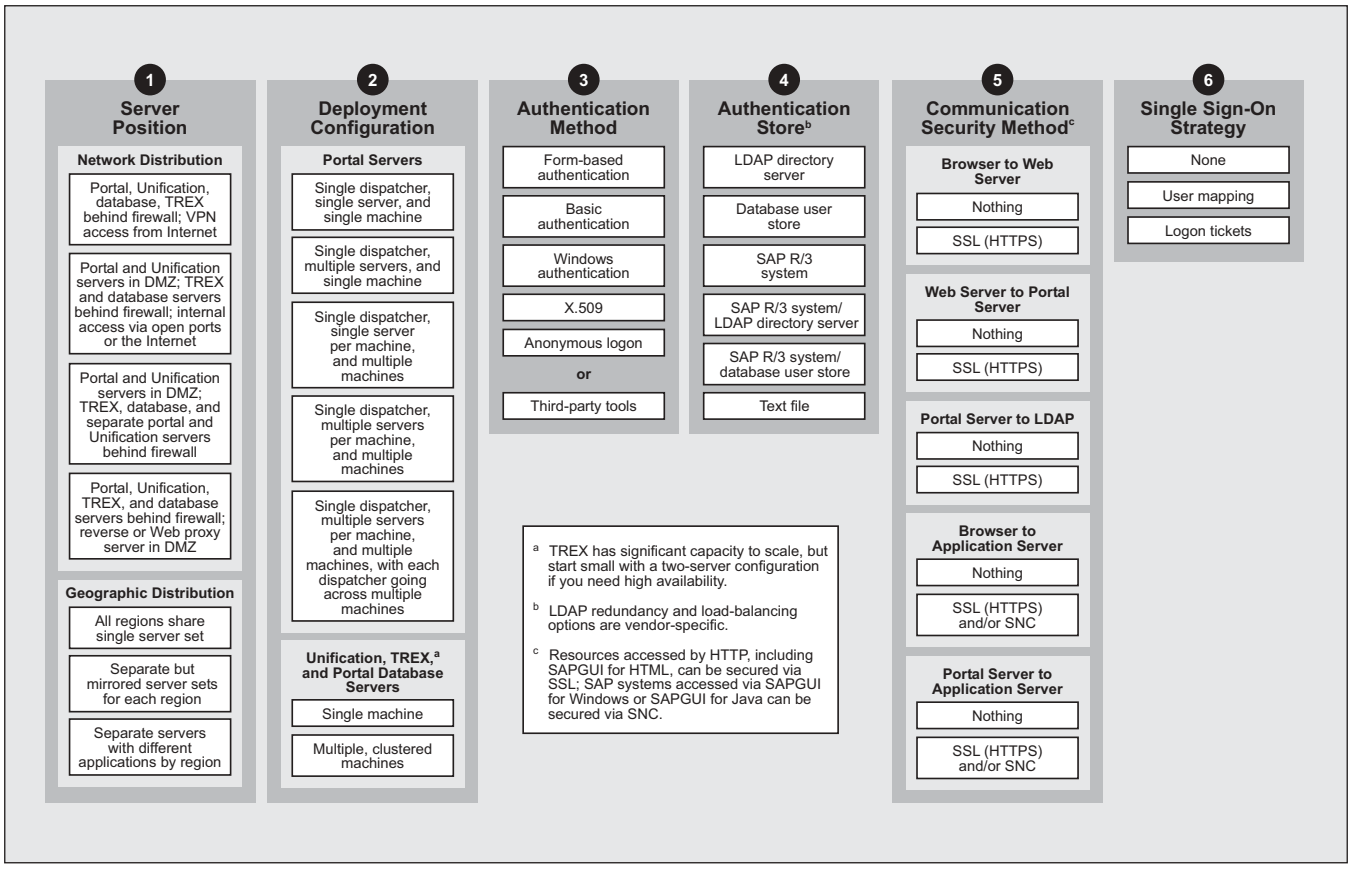
- An **intranet landscape**, which illustrates a basic "inward-facing" landscape designed for access by internal users, or external users connecting to your intranet via VPN[3]
- A **DMZ landscape**, which shows you how most network architects have traditionally designed "outward-facing," Internet-enabled landscapes[4]
- A **proxy landscape**, which shows you a simpler, more secure way to Internet-enable your portal

[2] I developed the simple implementation guide shown in Figure 1 to use on my projects and to orient client project teams new to SAP EP. Note that steps 3.0-4.4 are included for your reference only. These steps are not explicitly covered since they involve hardware implementation details that are beyond the scope of this article series.

[3] A virtual private network (VPN) involves installing client software and a software-based security key on employee desktop PCs and a VPN server in the DMZ (a Demilitarized Zone in this context is a protected network area between the Internet and your intranet). Employees can then use the VPN client software to establish a secure connection to your internal network.

[4] While many portal shops use this design, I do not recommend it; I'll explain why in the next section.

Figure 2 The Six Key Choices for Designing SAP Enterprise Portal Landscapes



✓ ***Note!***

While there are an infinite number of possible landscape designs, I developed the five model landscapes shown in this article to demonstrate key accessibility, security, and availability options. I recommend that you start by choosing one of the first three "base" designs (intranet, DMZ, or proxy), and then add security and availability features as you need them, since each added feature will tack on extra costs to the landscape.

✓ ***Note!***

If your portal is Internet-accessible, even if you provide internal (direct) access to your portal servers, make sure to give support personnel Internet access that bypasses the firewall, so they can see exactly what customers or vendors see when they call for technical support (e.g., so they can detect if your external firewall is malfunctioning).

by placing a reverse proxy or Web proxy server inside your DMZ

- A **high-availability landscape**, which demonstrates your options for high availability
- A **high-security landscape**, which shows you how to secure communications between each of the major landscape components

Throughout the article, a rough understanding of your requirements will be of great help to you when you set out to apply the technical options and sample landscapes described here to your company's unique needs (also keep in mind that platform requirements vary among the different portal components; see the sidebar on the next page for details). Over the next sections, with the five sample landscapes in mind, we'll look at the key options listed in Figure 2 by answering the following questions:

- Where will you position your servers?
- How will you configure your deployment?
- How will you authenticate users?
- Where will you store authentication information?
- How will you secure network communications?
- How will you implement single sign-on?

> ✓ ***Note!***
>
> *Although this article focuses primarily on developing portal landscapes for SAP EP 6.0, you can also apply the discussion to an SAP EP 5.0 installation if you take into account the architectural differences between 5.0 and 6.0, which were outlined in my previous article. You can also find detailed background information on the SAP EP architecture in the article "Integrating SAP Transactions, Reports, and Data into Your SAP Enterprise Portal — A Guided Tour of Your Options, Which to Use, and When" (SAP Professional Journal, July/August 2004) and in the SAP online help.*

Where Will You Position Your Servers?

As indicated in column ❶ of Figure 2, there are two aspects to consider when it comes to where to position your servers: network distribution and geographic distribution. The options for geographic distribution — share a single server set among all regions, use separate but mirrored servers for each region, or use separate servers with different applications for each region — are included in Figure 2 for completeness and are self-explanatory. The options for network distribution, however, are a bit more complicated and warrant further discussion:

1. Place the portal, Unification, TREX, and database servers behind the internal firewall, and use a VPN server in the DMZ for external access via the Internet.
2. Place the portal and Unification servers in the DMZ, place the TREX and database servers behind the internal firewall, and provide access via open ports or the Internet.
3. Place the portal and Unification servers in the DMZ, and place the TREX, database, and separate portal and Unification servers behind the internal firewall.
4. Place the portal, Unification, TREX, and database servers behind the internal firewall, and set up a reverse proxy or Web proxy server in the DMZ.

Option #1: Place the Portal, Unification, TREX, and Database Servers Behind the Internal Firewall, and Use a VPN Server in the DMZ for External Access via the Internet

This option is the obvious choice for internal, employee-only portals. All components — the portal server, the Unification server, the TREX server, the databases, and systems such as SAP R/3 and SAP BW — are placed behind the internal firewall for maximum security. Access to the portal server from the Internet is provided solely by VPN via HTTP or via HTTP secured by SSL (HTTPS), depending on how

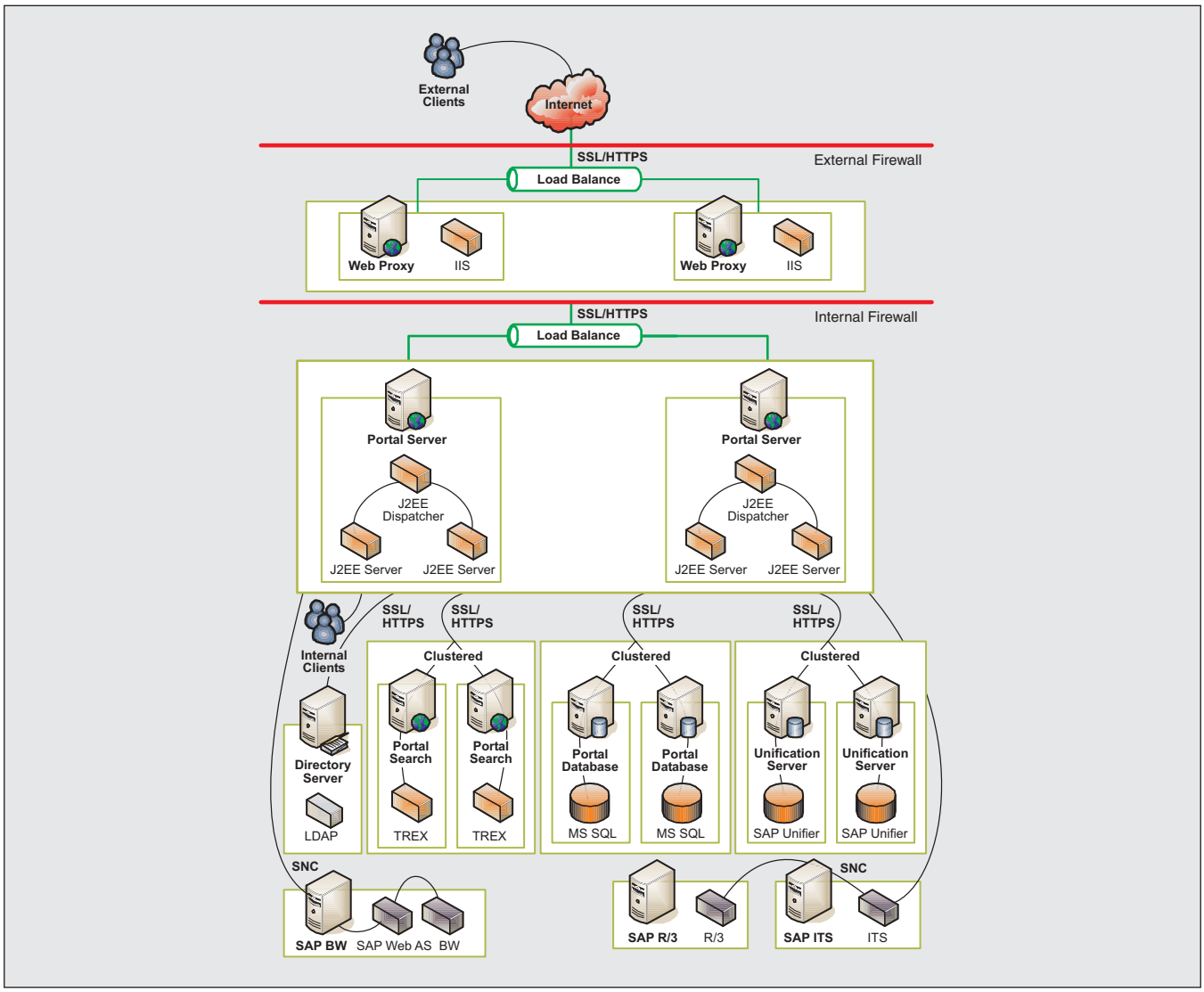
System Requirements for Major Portal Components

One thing that catches many SAP EP teams by surprise is that the various portal components have different platform requirements. The following table summarizes the key portal components and their platform requirements according to release as of this writing.

<table>
<tr><th>Component</th><th>SAP EP 5.0</th><th>SAP EP 6.0 SP1-SP2</th><th>SAP EP 6.0 SP3*</th></tr>
<tr><td>Portal Server</td><td>Requires a Microsoft Windows box running Microsoft IIS.</td><td colspan="2">Can run on Microsoft Windows 2000 and 2003, AIX, HP-UX, and Sun Solaris.</td></tr>
<tr><td colspan="4">The portal server includes an instance of the SAP J2EE Engine, which includes an HTTP daemon (a “Web server”) that enables it to receive and respond to HTTP requests from user Web browsers for personalization data and iView content.</td></tr>
<tr><td>Portal Database Server</td><td colspan="2">Runs on both Oracle and Microsoft SQL Server, either locally on the portal server or on a dedicated server.</td><td>Runs on Oracle, Microsoft SQL Server, and DB2/UDB, either locally on the portal server or on a dedicated server.</td></tr>
<tr><td colspan="4">The portal server stores role definitions, page-to-role relationships, and personalization data in a portal system database on the portal database server. User IDs, passwords, and contact and account information are stored in a user authentication store, which can be part of the portal system database on the portal database server, or located on a separate LDAP directory server or an SAP R/3 system.</td></tr>
<tr><td>Unification</td><td colspan="2">Runs on Microsoft Windows only with Microsoft IIS. Requires a Microsoft SQL Server running either locally or remotely.</td><td>Folded into the portal server.</td></tr>
<tr><td colspan="4">Unification is an optional, standalone component included with SAP EP. Unification’s main feature is Drag&Relate, which enables you to define hotspots on portal pages that users can drag and drop objects onto. SAP Unifiers are optional, add-in modules that construct the relationships required to “drag-enable” objects, or you can build the relationships yourself by defining a “unifier project.”</td></tr>
<tr><td>Knowledge Management & Collaboration</td><td colspan="3">Installed on the portal server. Database installed either on the portal server or on a database server.</td></tr>
<tr><td colspan="4">SAP Knowledge Management (SAP KM) is an optional component that provides a content management system for SAP EP. The Collaboration portion of SAP KM provides asynchronous (e.g., Collaboration Rooms) and synchronous (e.g., instant messaging) methods of communication. Searching and classification is provided by the Text Retrieval and Extraction (TREX) subcomponent (see below).</td></tr>
<tr><td>TREX</td><td colspan="3">Runs on both Microsoft Windows and Unix. Should be installed on a separate server from the portal.</td></tr>
<tr><td colspan="4">Text Retrieval and Extraction (TREX) is a subcomponent that serves as the search engine for the SAP EP content management system (SAP KM). It can also be used as the search engine for SAP Knowledge Warehouse (SAP KW) and SAP Customer Relationship Management (SAP CRM).</td></tr>
<tr><td colspan="4">* SAP Enterprise Portal 6.0 SP3 has been renamed “SAP Enterprise Portal running on SAP Web Application Server 6.40” to emphasize SAP Web AS as its underlying platform.</td></tr>
</table>

The bottom line is that if you want to run all of these systems on one box, perhaps as a development system, you’ll need to run Microsoft Windows and Microsoft SQL Server, since that is the only platform supported by Unification up to and including SAP EP 6.0 SP2. For production environments, SAP strongly recommends (and I agree) that you run Unification and TREX on their own boxes, since they are highly resource-intensive.

Figure 3 An Inward-Facing "Intranet Landscape," Internet-Accessible via VPN Only



you choose to secure your network communications (more on this later in the article); behind the firewall the portal server can then access backend systems via HTTP, JDBC, and LDAP, for example. Since portal-related network traffic is restricted to the internal network, the connections between systems normally don't need to be secured, except if you consider parts, or your entire network, insecure.[5] VPN is the cheapest, safest, and most flexible way to let employees access the portal over the Internet. **Figure 3** shows an example intranet landscape that uses the VPN option.

> ✓ ***Note!***
>
> *The portal and TREX servers in Figure 3 contain the SAP Internet Transaction Server (ITS) components WGate and AGate, which Web-enable SAP R/3 transactions. For more on ITS and its role in SAP EP, see the sidebar on the next page.*

[5] In my previous article, I gave an example of a company that chose to secure communications over a line between two buildings that might potentially be tapped.

Option #2: Place the Portal and Unification Servers in the DMZ, Place the TREX and Database Servers Behind the Internal Firewall, and Enable Access via Open Ports or the Internet

This approach offers both internal and external access to a single set of portal servers. While Web landscape designers have traditionally placed Web-based servers in the DMZ to Web-enable their systems, in my opinion, this is not the best approach for portal landscapes, for the following reasons:

- The DMZ is less secure than your intranet, and is more easily infiltrated. Placing middleware components here exposes them to unnecessary risk.
- A large number of ports need to be opened on the

Positioning Your ITS Servers in a Portal Landscape

First introduced in the late 1990s, SAP Internet Transaction Server (ITS) Web-enables SAP systems by providing instant access to SAP transactions and reports with its SAPGUI for HTML feature.* It is a predecessor to SAP Web AS, and is required for many business packages since it includes features not yet integrated into SAP Web AS. For example, as of this writing, the Employee Self-Service (ESS) business package and SAP Unifiers require ITS in order to access transactions from the Web. Since ITS has been around for a while, you may already have an ITS server in your landscape — if you've Web-enabled any SAP R/3 applications, or if you have or had SAP BW 2.x or SAP Workplace, for example.

ITS has historically been released as a standalone server product consisting of two main components: a Web server gateway (WGate) component and an application server gateway (AGate) component. The WGate is basically a plug-in that is available for Microsoft IIS, Apache, and Sun ONE Web servers. It can run on any platform these Web servers can run on. The AGate is a full application server, and is only available for Microsoft Windows and Linux systems. The latest standalone release of ITS is version 6.20, and can be downloaded free of charge from **http://service.sap.com**. For complete details on platform availability, see SAP Note 325616.

If you don't already have ITS, you have three main options for setting it up:

- **Single, standalone installation:** Install the WGate and AGate components on a Microsoft Windows or Linux server running Microsoft IIS, Apache, or Sun ONE (Figure 4 in the article shows this type of installation).
- **Dual, standalone installation:** Install the WGate component on a server running Microsoft IIS, Apache, or Sun ONE, and the AGate component on a separate Microsoft Windows or Linux box (this type of installation is not shown in the article).
- **Consolidated:** Install the WGate component on the portal server, and the AGate component on a shared TREX or Unification server running on Microsoft Windows or Linux (Figure 5 in the article shows this type of installation). Since SAP EP 5.0 requires Microsoft IIS, the WGate can be installed directly on the SAP EP 5.0 portal server. SAP EP 6.0 uses an SAP proprietary Web server that the

* SAPGUI for HTML dynamically converts ABAP screens into HTML so they can be viewed with a Web browser. The pages look remarkably similar to SAPGUI for Windows.

(continued on next page)

(continued from previous page)

WGate cannot use, however, so to run the WGate on your portal server you must run a Microsoft IIS, Apache, or Sun ONE Web server in parallel. Since the AGate is resource-intensive, do not place the AGate on your portal server.

Note that SAP has recently merged the ITS SAPGUI for HTML feature into the SAP Web AS 6.40 ABAP kernel as an ICM service.** Beware, however! The new, integrated ITS in SAP Web AS 6.40 cannot be used to access remote SAP systems — it can only access SAP transactions on the local ABAP system (if installed). So, unfortunately, even though SAP Web AS 6.40 will underlie SAP EP 6.0 SP3, you'll still need a standalone ITS server like ITS 6.20 to access your backend SAP BW and SAP R/3 systems — i.e., unless you upgrade each of the servers running these systems to SAP Web AS 6.40 (note that a pure kernel upgrade to SAP Web AS 6.40 will not install the integrated ITS, however, as explained in SAP Note 709038). For more information on the new integrated ITS in SAP Web AS 6.40, visit **http://service.sap.com/sapgui** and click on the SAPGUI for HTML link.

** The Internet Communication Manager (ICM) is essentially the HTTP daemon within SAP Web AS (refer back to the sidebar on page 7). For more information, go to *SAP Web Application Server → Client/Server Technology (BC-CST) → Architecture of the SAP Web Application Server → SAP Web Application Server Components → Internet Communication Manager (ICM)* in the SAP Library.

internal firewall to enable the portal server to access backend systems via HTTP, JDBC, and LDAP, for example, as shown in **Figure 4**

- Adding access for your internal users will require you to either open ports on the internal firewall (which is undesirable from a security perspective), or enable access via the Internet (which is undesirable from both a security and performance perspective, since data needs to go "out" in order to come back "in").

- Maintaining portal and Unification servers in a DMZ is much more difficult, because network access is restricted and special security measures must be implemented during upgrades.

- Placing a firewall between the application server and database servers can seriously degrade system performance.

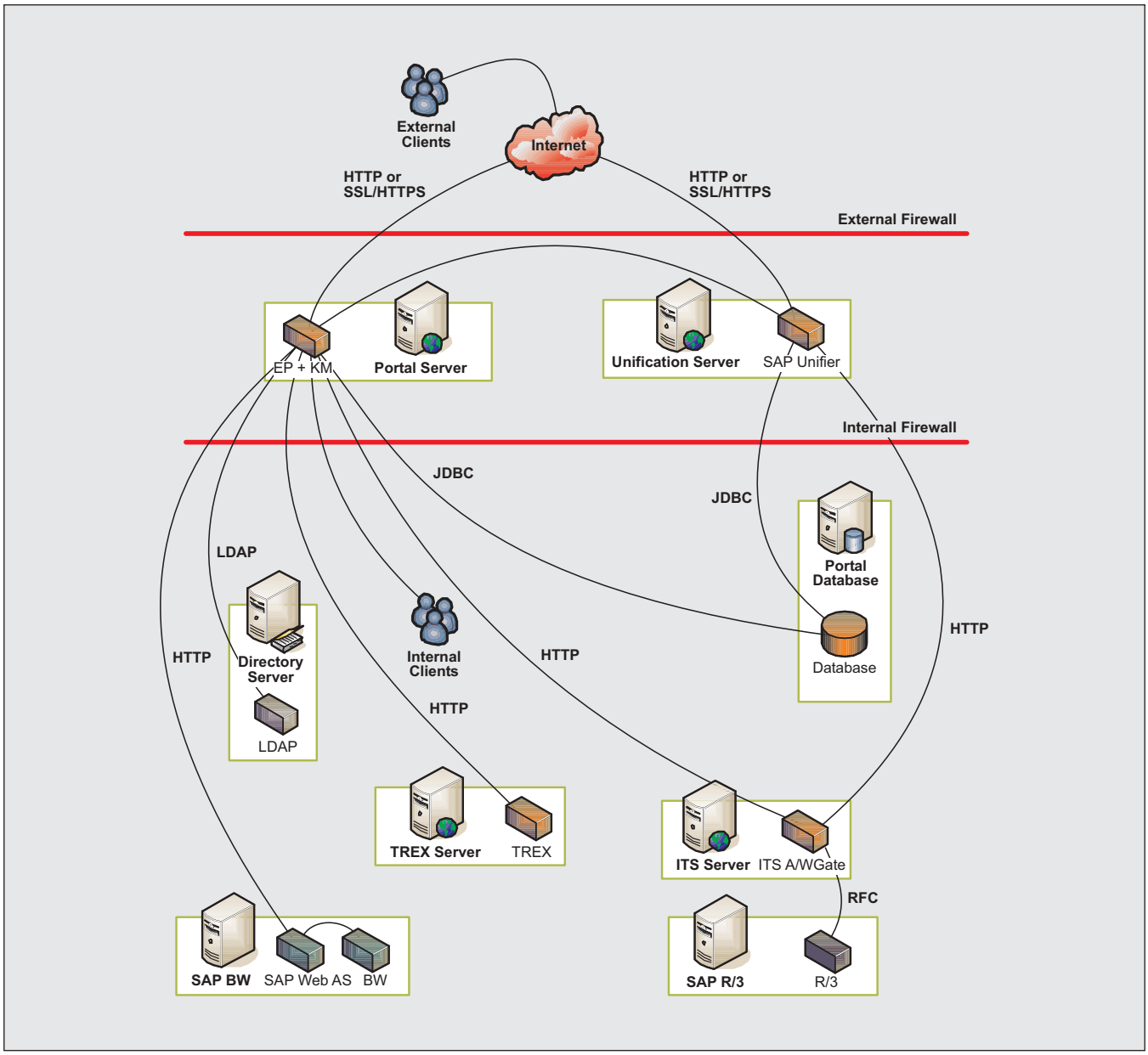
For these reasons, if you need Internet accessibility, I recommend that you use the Web proxy or reverse proxy approach (Option #4, which we'll discuss shortly) if at all possible.

If you plan to use X.509 digital certificates, however, the proxy option may not be available to you. In this type of scenario, the proxy gateway is the end point of the SSL client authentication rather than the portal server (more on this later), so if you would like to use X.509 digital certificates, you'll need to revert to Option #2, which is shown in the DMZ landscape illustrated in Figure 4.

✓ ***Note!***

If you open ports on your internal firewall for access to the DMZ, be sure to only open them in the intranet-to-DMZ direction (i.e., for requests initiated from the intranet and sent to the DMZ). This is a setting on most firewalls, and is more secure than the default bi-directional setting.

Figure 4 An Outward-Facing "DMZ Landscape" with Portal and Unification Servers in the DMZ



Option #3: Place the Portal and Unification Servers in the DMZ, and Place the TREX, Database, and Separate Portal and Unification Servers Behind the Internal Firewall

If you choose Option #2 for external access (where the portal and Unification servers are in the DMZ), and need to offer access to internal users, setting up a separate set of servers for internal users is a less risky alternative to opening ports in the firewall or enabling access over the Internet.[6] There are four key benefits to this approach over firewall port or Internet access:

- You don't have to worry about the security risks

[6] While I haven't illustrated this landscape explicitly, it is essentially a combination of Figures 3 and 4.

involved in data traveling from your internal system out to the Internet and back again.

- Internal users will benefit from improved response times (since they are accessing an internal portal, traffic doesn't repeatedly cross the firewall).
- You can choose to deploy high-security iViews (or applications) to the internal portal only, since internal communication is more secure.
- You can geographically distribute the internal portal servers to optimize performance (most company DMZs are located in a single region, preventing you from geographically distributing the portal servers that are located there).

> ✓ ***Tip***
>
> *Be aware that running separate sets of applications on different portal machines instead of making them mirror images of one another will make deploying and maintaining applications much more difficult; you'll have to track which applications belong on which servers. Also keep in mind that SAP EP automatically mirrors applications deployed on one server to all servers in the same cluster, so you can only cluster portal servers that are mirror images of each other. In addition, any load-balancing portal servers should be identical to the servers they are balancing in terms of the operating system in use and any installed software.*

Overall, this approach is more strategic than Option #2, and involves less risk since only external users are using the external servers. Consider this option when you need maximum performance, or if you can't use the proxy option (Option #4, discussed next) for some reason (if you plan to use X.509 digital certificates, for example).

Option #4: Place the Portal, Unification, TREX, and Database Servers Behind the Internal Firewall, and Set Up a Reverse Proxy or Web Proxy Server in the DMZ

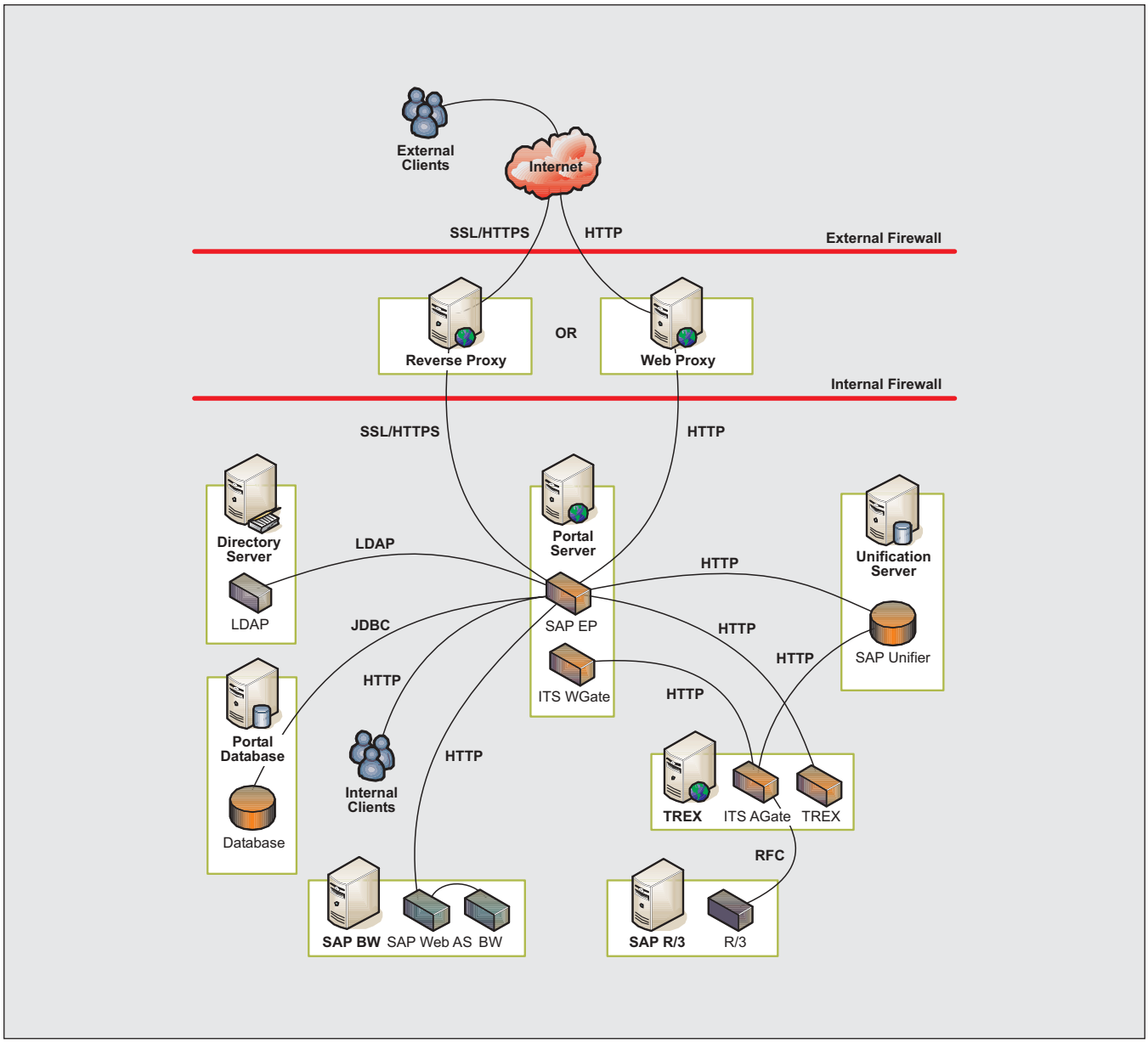
Figure 5 demonstrates this simple, flexible, secure configuration. As shown in the diagram, there are two types of proxies you can use to enable external access to your portal, Unification, and other servers:

- **Reverse proxy:** A reverse proxy (also referred to as a "proxy gateway") is a dedicated hardware-based proxy — for example, an Apache or Microsoft IIS server — that receives all HTTP and HTTPS requests from the Internet on behalf of the application servers. It acts as a generic intermediary between the Internet and your intranet, and securely forwards requests based on a set of rules you define. It protects the portal server by allowing it to be placed behind the firewall in the secure network, and by acting as a mediator between portal clients and the portal server. Previously, companies had to place actual application servers in the DMZ (as with Options #2 and #3) in order to achieve this buffering.

> ✓ ***Note!***
>
> *The reverse proxy can distinguish between requests coming from the Internet and those coming from the intranet, so you can apply separate filtering rules if needed.*

In a typical reverse proxy landscape, the reverse proxy is configured for SSL, not the portal server. The proxy encrypts client requests before forwarding them to the portal server and decrypts them before sending the response back to the client. Client authentication with X.509 digital certificates is not supported with a reverse proxy configuration, because in this case the proxy

Figure 5 A Dual-Facing "Proxy Landscape" with Internet Access via a Reverse or Web Proxy



gateway rather than the portal server is the end point of the SSL client authentication. Reverse proxies are ideal if you have multiple HTTP servers that need a single point of access to the outside world. The disadvantage is that if the reverse proxy goes down, and there is no failover, then all of your HTTP servers would become unavailable in one fell swoop.

- **Web proxy (SAP EP 6.0 only):** This is the quickest and least costly option for most companies. Instead of using a dedicated hardware-based proxy, you instead reuse the Apache or Microsoft IIS Web server hosting your company's Internet site in the DMZ by installing the necessary software on that existing server. To add Web proxy capability to a standard Microsoft IIS or Apache server, download

the Apache or IIS plug-in and filter for your server from **http://service.sap.com**, install them on the server, and then modify the filter's XML configuration file so that it forwards Internet URL requests to the appropriate middleware server (e.g., the portal server, Unification server, BW Web server, etc.) — i.e., specify the URL path, port number, etc. for each middleware server. By default, the Web proxy is configured for HTTP. To configure the Web proxy to communicate with the backend portal server via SSL, you'll need to install and configure the SAP Cryptographic Library.[7]

> ✓ ***Note!***
>
> *If the DMZ-based Web proxy server is configured for SSL, incoming HTTPS requests are decrypted at the Web proxy server. If you want the Web proxy filter to re-encrypt the requests before sending them to the backend systems, you'll need to install and configure the SAP Cryptographic Library. I'll discuss SSL more in the section "How Will You Secure Network Communications?" later in the article. See the SAP EP 6.0 security guide for more information on securing DMZ Web-server-to-portal-server communications.*

> ✓ ***Tip***
>
> *Unless you find it strategic to isolate your portal communications, you don't need to set up a dedicated virtual Web site on your existing Microsoft IIS or Apache box for the portal. Installing the Web proxy filter on an existing Web site will keep administration simple, and give users a single address/port to use.*

[7] For information on how to install the SAP Cryptographic Library, in the SAP Library navigate to *SAP Web Application Server → Security (BC-SEC) → SAP Web Application Server Security → Using the Secure Sockets Layer Protocol → Installing the SAP Cryptographic Library on the SAP Web AS.*

> ✓ ***Tip***
>
> *Make sure to configure your proxy XML file to forward requests to all backend systems your users will need to access directly. See the section "How Will You Secure Network Communications?" for more on the various paths portal data take.*

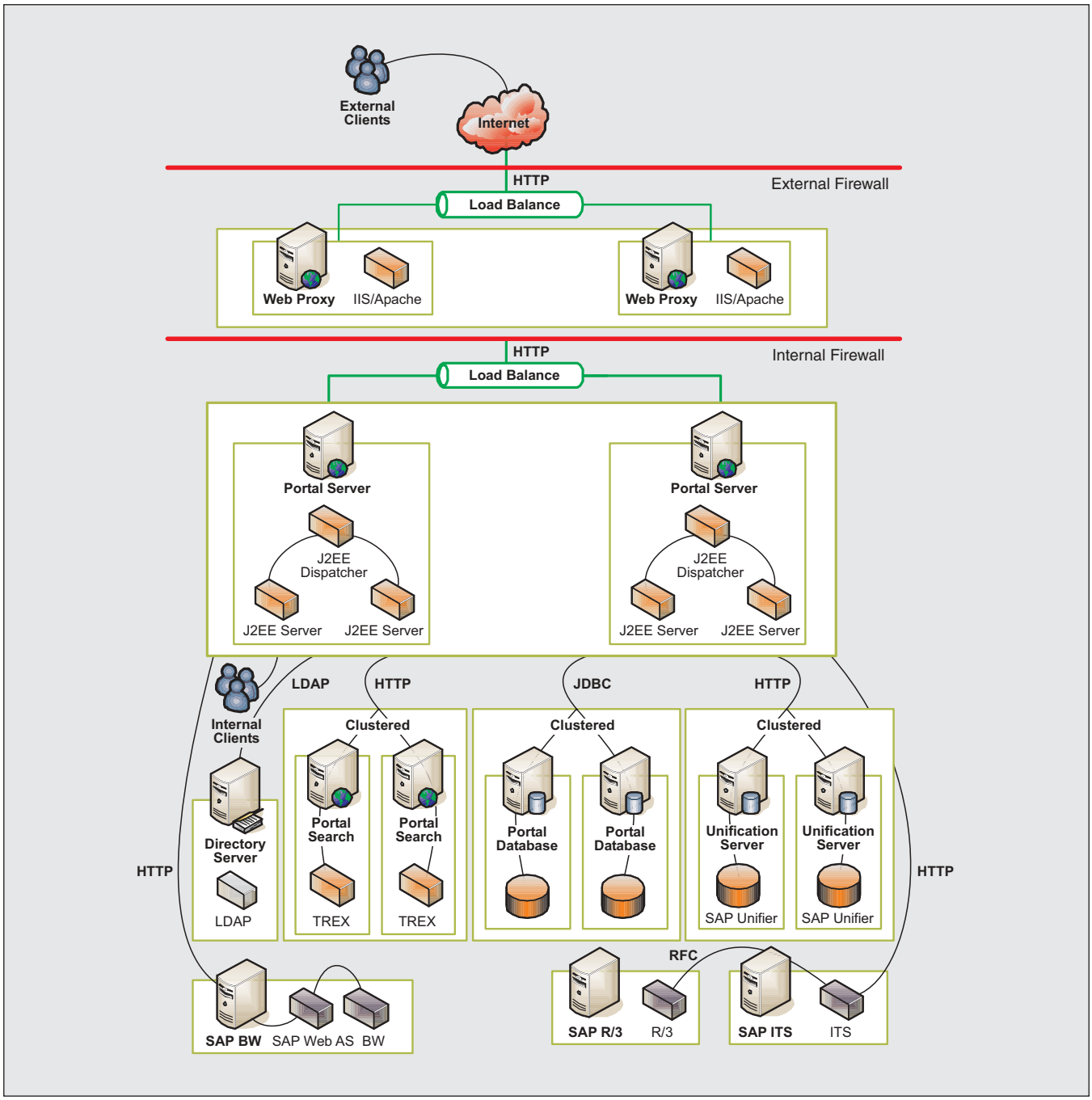
Keep in mind that while the Microsoft IIS/Apache Web proxy provides access to the portal server, any externally facing portal architecture also needs to make all of the integrated backend applications available to the end user. This means that all backend applications must also be directly available from the outside. The advantage of Web proxies over reverse proxies is that they are easier to set up and also offer single sign-on to non-SAP applications. The disadvantage is that they are not necessarily configured for high security.

Regardless of whether you choose a reverse proxy or a Web proxy, SAP recommends that you consider the proxy approach first if you need Internet accessibility, and based on my own experience, I agree — it's quick, safe, and maximizes your portal resources since all users share a central set of servers. It also provides companies that start with an inward-facing portal an inexpensive, flexible way to add Internet accessibility after the initial implementation.

How Will You Configure Your Deployment?

SAP EP components offer powerful redundancy and load-balancing options. Providing redundancy does not just provide failover protection, it also allows the solution to scale as your needs grow. As shown in column ❷ in Figure 2, we'll focus mainly on the portal server here since it's the most critical component and offers the most options (see the sidebar on page 19 for a brief discussion of the high availability options for other components).

Figure 6 A Dual-Facing "High-Availability" Landscape with Redundant, Load-Balanced Servers



The two most important portal server components with respect to availability are the J2EE dispatcher and J2EE server, both of which are part of the SAP J2EE Engine that underlies SAP EP 6.0. The J2EE dispatcher includes an HTTP daemon (a "Web server") that receives requests from the client and forwards them to the J2EE server. Architectures that use a proxy as a mediator for load-balanced portals (see **Figure 6**) require a proxy for each J2EE dispatcher.

The following are the basic deployment configurations available for the portal server, in order of increasing cost (see **Figure 7** for a summary):

1. Install a single J2EE dispatcher and a single J2EE server on a single machine.
2. Install multiple J2EE servers that receive requests from a single J2EE dispatcher on a single machine.
3. Install a single J2EE dispatcher and a single J2EE server per machine, then load balance the machines using a software- or hardware-based load balancer.
4. Install a single J2EE dispatcher and multiple J2EE servers per machine, then load balance the machines using a software- or hardware-based load balancer.
5. Same as Option #4, except that the J2EE dispatcher on each machine communicates with the J2EE servers on every other machine.

Option #1: Install a Single J2EE Dispatcher and a Single J2EE Server on a Single Machine

This is the most common configuration for a single-box installation, and does not implement any software or hardware failover protection. It is the de facto choice for development environments, and many quality assurance environments as well. Consider using one of the other options (like Option #2) for your quality assurance portal, however, to familiarize yourself with the operation and testing of the portal's load-balancing and failover features. This configuration is not appropriate for production installations due to the lack of any redundancy for the portal server — your portal will not work if the portal server goes down, and downtime is costly.

Option #2: Install Multiple J2EE Servers That Receive Requests from a Single J2EE Dispatcher on a Single Machine

This configuration provides a basic level of software redundancy, and was not available with SAP EP 5.0. Once you complete the initial portal installation, you can add J2EE server instances on the same machine via the SAP J2EE Engine configuration tool.

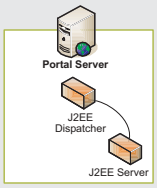
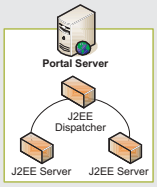
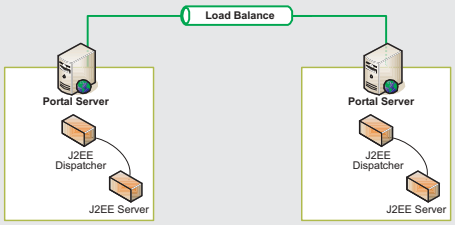
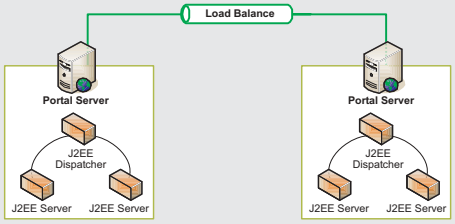
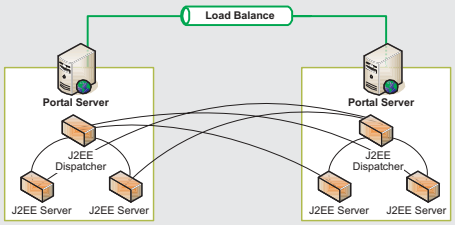
> ✓ ***Tip***
>
> *In order to meet the scaling guidelines of the portal, I recommend that you install no more than one J2EE server instance for every 2 CPUs on the portal machine. For example, if you want to run two J2EE servers on a single machine, it should be a quad processor CPU.*

Because the portal server has two J2EE server installations with this configuration, it protects you from any software glitches encountered on any single J2EE server. Of the portal's components, the J2EE server is the most likely component to fail since it hosts key operational components like the Page Builder and the iView runtime. Remember that this configuration still leaves you exposed to J2EE dispatcher failures, operating system crashes, and hardware failures, however. If someone turns off the power on that machine, for example, your portal will go down.

> ✓ ***Note!***
>
> *There can only be one J2EE dispatcher node per portal instance. This J2EE dispatcher node coordinates communication between the J2EE server node (or nodes) and the client. In order to have J2EE dispatcher redundancy, multiple portal machines have to be installed and load balanced.*

Nevertheless, this is a reasonable configuration for low-volume portals where availability isn't critical — for example, a training system, a low-cost initial pilot

Figure 7 Summary of Portal Server Deployment Configurations

Scenario	Features	When to Choose
Option #1: 	• Single J2EE dispatcher • Single J2EE server • Single machine	• Most basic configuration, appropriate for DEV installations • Can also be used for QA installations if higher capacity machines are not available
Option #2: 	• Single J2EE dispatcher • Multiple J2EE servers • Single machine	• Appropriate for QA and training systems • Appropriate for low-volume, point-solution portals running non-mission-critical applications
Option #3: 	• Single J2EE dispatcher • Single J2EE server per machine • Multiple machines	• Basic configuration for load-balanced production environments • Ideal configuration if machines larger than 2 CPUs are not available
Option #4: 	• Single J2EE dispatcher • Multiple J2EE servers per machine • Multiple machines	• Appropriate for high-availability production environments • Enables geographic distribution of machines • Preferred solution if larger machines are available since each J2EE server requires an additional 2 CPUs
Option #5: 	• Single J2EE dispatcher • Multiple J2EE servers per machine • Multiple machines • J2EE dispatchers configured to dispatch across machines	• Appropriate for high-availability production environments • Choose only when machines are geographically close to each other

portal, or a point-solution portal running non-mission-critical applications. Since the only downside to this option is the small amount of extra effort to set up the additional J2EE servers, I strongly recommend this over Option #1 when you have the requisite CPU capacity.

Option #3: Install a Single J2EE Dispatcher and a Single J2EE Server Per Machine, Then Load Balance the Machines Using a Software- or Hardware-Based Load Balancer

This is the most common configuration for load-balanced production systems, and was the de facto load-balancing solution with SAP EP 5.0. All users are assigned to a certain portal server on initial logon, using a sticky round-robin load-balancing approach. If a portal server fails, all users on the failed server are automatically switched over to one of the other available servers on the next browser refresh, without any apparent service disruption. Since load-balanced portal servers are mirror images of each other — i.e., contain the same applications — all applications on the failed server remain available to the end user. You can use the Network Load Balancer (NLB) that comes with Microsoft Windows Advanced Server as a software-based load balancer, or you can use Cisco 11509 as a hardware-based load balancer. The one you choose depends on your preference and your needs; while NLB comes free with Microsoft Windows Advanced Server, a hardware-based solution can load balance two or more middleware applications.

Once a single portal instance has been installed and is functioning, additional portal instances can be installed on additional server machines using the SAPinst tool. Each additional portal instance is installed and connected to the initial portal instance as part of the same J2EE cluster, and all deployed applications are automatically synchronized across all the server machines in that cluster. Figure 6 shows this configuration, which is preferable for production environments where the machines are no bigger than 2 CPUs each.

Option #4: Install a Single J2EE Dispatcher and Multiple J2EE Servers Per Machine, Then Load Balance the Machines Using a Software- or Hardware-Based Load Balancer

This option is essentially a combination of Option #2 and Option #3 in that it provides either software-level or hardware-level load balancing. This is a more complicated installation, but once configured provides a high level of availability and failover. Again, the Network Load Balancer provided with Microsoft Windows Advanced Server can be used as a software-based load balancer, or Cisco 11509 can be used as a hardware-based load balancer. This configuration is appropriate for a high-availability, mission-critical production environment that is running on quad processor machines. It can also be used for geographically dispersing your portal servers.

Option #5: Same As Option #4, Except That the J2EE Dispatcher on Each Machine Communicates with the J2EE Servers on Every Other Machine

This configuration provides the highest availability for a portal solution. Even if all J2EE servers in Machine A are out of service, the J2EE dispatcher of Machine A can communicate with J2EE servers in Machine B without any service disruption to the users connected to Machine A. If the J2EE dispatcher in Machine A fails, however, then all users on Machine A are switched over to the J2EE dispatcher of Machine B, which will have access to the J2EE servers of both Machines A and B. This deployment is appropriate for high-availability production environments where the portal servers are not geographically dispersed.

As you've seen, SAP EP 6.0 provides numerous options for incorporating redundancy for the portal server (refer back to Figure 7 for a summary). These options are available for each portal component implementation (see the sidebar on the next page) and should be weighed carefully before making a decision. Your final choice will depend on the required availability of the server, the geographic distribution of your machines, and the server capacity.

Configuring Other Portal Components for High Availability

When designing your landscape, it's important to consider all the systems your portal will rely on, not just the portal server itself:

- **Unification:** Unlike the portal server, Unification in SAP EP 6.0 SP2 requires a dedicated Microsoft Windows machine equipped with a Microsoft IIS Web server, plus a Microsoft SQL Server database installed either on the same machine or on another machine. Overall, Unification has a much simpler architecture than the portal server. SAP includes a simple Unifier Load Balancing Configuration tool with the Unification component for load balancing your Unification database, SAP R/3 system, or SAP BW system. It's easy to use, and I highly recommend it if you set up redundant Unification servers. There are some considerations, however. In order to have a load-balanced Unification environment, the configuration tool must be installed and running on two or more machines. These installations must be identical, and they must not be on the portal server machine. One reason is because the tool essentially designates one of the machines as the shared location for the load-balancing environment, and provides a centralized persistence layer for all the defined unifier projects.* Note that in SAP EP 6.0 SP3, Unification has been merged into the portal server platform, so that it now shares the redundancy and load-balancing options of the portal server.

- **TREX:** TREX can be installed in either a Microsoft Windows or Unix environment. It is extremely resource-intensive and it is strongly recommended that you install it on its own server. TREX has a highly sophisticated architecture with a half dozen separate logical components, each with its own distribution criteria. Unless searching is a critical feature for you, TREX should be installed on a single server initially, and then scaled as your needs grow. While most teams find a dual-server TREX installation sufficiently powerful for their entire enterprise, TREX can technically be scaled to Amazon.com levels of performance.

- **Portal database server:** For the database server, an active/passive cluster is generally recommended. If a cluster configuration is not possible, you should consider setting up a hot-swappable server for the portal database. If your organization has an existing storage array network (SAN), then it should be leveraged as well. The specific options and procedures will depend on your database, so consult your administrator or the appropriate database documentation.

- **Directory server:** For a directory server, the domain controller can be replicated to provide backup for a Microsoft Windows-based system. As with the database server, the specific options and procedures available will depend on your specific directory server vendor.

- **Load balancer:** The load balancer hardware must be duplicated at the hardware level. This allows for one or more load balancers to receive the request to forward on to a portal server. This is a single point of failure many teams forget!

- **Backend systems:** Remember that SAP EP relies heavily on content from backend systems like SAP R/3, ITS, legacy systems, and SQL databases, so don't forget to include the availability of backend systems in your implementation plan!

---

* Unifier projects define the relationships that "drag-enable" objects (see the sidebar on page 7).

> ✓ ***Note!***
>
> *If company employees are in two separate locations with their own data centers, and both locations are on the same network cloud and connected via a high-speed network, they could each have a dedicated portal server that is load balanced with the other. This could provide redundancy not only for the portal machine, but also in case one of the data centers goes offline for some reason.*

How Will You Authenticate Users?

User authentication for your portal can be performed by either a third-party product like Netegrity SiteMinder, Entrust GetAccess, or Novell iChain, or by the portal itself. If you are already using a third-party product, you may want to continue using it for your portal if it is strategic for you to do so, or if you need functionality that is not provided by SAP EP, like token cards. Keep in mind, however, that if you use a third-party authentication product, users will be redirected to the third-party login page rather than your standard portal login page, and it can be costly to change your authentication strategy later, so weigh the decision carefully and consider your future needs. In my experience, most companies choose to have SAP EP perform user authentication if they do not have a third-party authentication package already in use.

If you choose SAP EP for user authentication, you must choose from one or more of the following methods (see column ❸ in Figure 2), which are configured within the user administration role:

1. Form-based authentication
2. Basic authentication
3. Windows authentication (a.k.a. NTLM[8] authentication)
4. Verification using X.509 digital certificates

Option #1: Form-Based Authentication

Form-based login is the traditional login method — the portal presents a Web page for login when users first try to access a portal resource, or after their session has exceeded the inactivity threshold.[9] This is the default authentication scheme for the portal.

Since passwords are normally passed over the network with clear text, however, you should strongly consider securing the network traffic using IPSec[10] or another similar network security protocol. Form-based login is the preferred option for most portals since it is easy to implement and does not require a Microsoft IIS Web server. If you need to, you can also customize the code behind the logon page so that the design reflects your company branding — simply modify the JavaServer Pages (JSP) file that determines the page's look and feel.

Form-based authentication is often used in combination with other authentication methods, including certificate-based authentication and self-registration.

Option #2: Basic Authentication

With basic authentication, the portal instructs the browser to prompt for a user ID and password with a dialog box.[11] The browser submits the user ID and password to the portal, and if valid, delivers the requested page. You can activate this model via a setting within the properties of the Microsoft IIS Web server.

There are three key considerations with basic authentication, however:

[8] The Windows NT LAN Manager (NTLM) is a proprietary Microsoft Windows protocol for authenticating users and machines and enabling single sign-on in Windows environments.

[9] The user session timeout setting can be adjusted in the user administration console.

[10] IP Security (IPSec) is a set of protocols developed by the IETF standards organization to support secure data exchange at the IP layer.

[11] You've seen this dialog if you've ever logged into **http://service.sap.com**.

- It requires a Microsoft IIS Web server.
- By default, basic authentication transmits passwords in clear text. If you choose this option, enable encryption by configuring the portal server for SSL for added security.
- There is no login Web page on which to put links to frequently asked questions, password reset instructions, anonymous login, etc. A workaround is to place these on the "invalid" password page that appears if the user authentication fails, but this is not as user-friendly.

For these reasons, basic authentication is primarily used in conjunction with Windows authentication (Option #3, discussed next), which suppresses the login dialog entirely.

Option #3: Windows Authentication (a.k.a. NTLM Authentication)

Used in combination with basic authentication, Windows authentication lets Microsoft Windows users who explicitly log on to their systems[12] to bypass the portal login dialog. This requires the Windows authentication user ID to be the same as the user ID in the portal's authentication data store. Non-Windows users, or those who fail authentication, will receive the standard prompt. Selecting NTLM is an option within the user administration console of SAP EP.

When considering Windows authentication, keep in mind:

- This option is only available for portal servers on a Microsoft Windows machine running on Microsoft IIS. Even though SAP EP 6.0 would be using its own Web server, Microsoft IIS would be used to configure the NTLM authentication.
- This option is not appropriate for users with shared workstations! If *any* of your users share workstations, you cannot use this model, because once desktop authentication is completed, the portal comes up without further security clearance.

Option #4: Verification Using X.509 Digital Certificates

In order to use X.509 digital certificate authentication, all users must have obtained a certificate based on a public key infrastructure (PKI) and imported it into their Web browsers. The PKI can either be self-generated or obtained from an external trust center such as VeriSign. Finally, the portal server must be configured to trust and accept the certification authority (CA) that issues the certificates. This approach provides a higher level of security by requiring SSL configuration between the client and the portal server.

While this option is ideal for situations where high-level security is required, remember that you cannot use X.509 digital certificates if you are using a proxy (because in this case the proxy is the authentication end point instead of the portal server).

Option #5: Anonymous/Guest Login

SAP EP 6.0 offers a self-registration feature whereby anonymous users can self-register and assign themselves to a "company" identity predefined by the portal administrator. Pending review and approval by an IT or customer administrator (depending on the configuration in the portal administration tool), self-registered users who assign themselves to a company are granted a default "guest" profile and are recorded in the portal database. If the registration is approved, the user is granted the roles associated with that company. If rejected, the user simply retains a guest status. Self-registered users who do not assign themselves to a company retain guest status indefinitely. This option is useful for situations where you have an external portal that provides self-service options such as those provided by Amazon.com.

[12] Microsoft Windows systems like Windows 98, ME, and 2000 support an anonymous logon capability. Ask your Windows administrator which system has been configured on your users' PCs.

By default, anonymous login is active as of SAP EP 6.0 SP2, and the self-registration link is available on the logon screen upon initial installation. If you do not want anonymous users logging into the portal, you can deactivate this feature in the portal system administration console. Self-registration can only be enabled if you use the form-based login approach.

Of the five authentication options discussed here, form-based login is the most common form of authentication, especially for companies that also permit self-registration. If the portal has been set up with X.509 certificate authentication, for example, guest users would still use form-based login to sign in.

SAP EP provides a good spectrum of delivery authentication options. Form-based login provides the default functionality for both HTTP and HTTPS portals. If integrated Windows authentication is required, then basic authentication can be used in conjunction with NTLM as long as a Microsoft IIS Web server is available. Certificate-based authentication is also provided with X.509 digital certificates for high security as long as an external trust center has been contracted.

Where Will You Store Authentication Information?

If you decide to use SAP EP for user authentication, rather than a third-party product, you'll need to decide where the portal should retrieve and store user authentication information.

The portal user authentication store contains numerous fields of data about each user, mainly the user ID and password. Other information can include contact information and user ID expiration date. This information can also be updated by the administrator, or by the user if they have access to their profile. Unlike SAP EP 5.0, which used a single LDAP server for all authentication data, SAP EP 6.0 lets you use one or more of the following as your authentication store (see column ❹ in Figure 2):

- **An LDAP directory server:** Supported directory servers include Microsoft Active Directory Server (ADS), Novell e-Directory, Siemens DirX, and Sun ONE. Other directory servers may also be compatible, but are not officially supported by SAP. For example, one of my clients had an implementation of the IBM directory server that worked with SAP EP. Most companies with an existing directory server loaded with user information choose this option, as there is usually at least one existing LDAP directory on the server, so try to locate one before setting up a new one. You can also allay your LDAP administrator's fears by telling him or her that, unlike SAP EP 5.0, SAP EP 6.0 only needs read access to the LDAP directory. You may have already noticed that the sample landscapes in the article use an LDAP directory server for user authentication information.

> ✓ ***Tip***
>
> *You may already have a directory server and not know it! If you use Microsoft Exchange for email, ask your LAN administrator if your Exchange system uses an LDAP server.*

- **The portal system database:** With this option, the portal stores user data within the portal system database (in a separate table from the configuration information). This is an easy, low-cost option that's appropriate when you don't have an existing LDAP server and don't want to invest in setting one up. It also removes the portal's dependence on an external solution to provide user authorization.

- **An SAP R/3 system:** If your SAP R/3 system is based on SAP Web AS 6.20 or higher, you can use it as your portal user authentication data store. This is a great option for companies with an extensive SAP R/3 user population to roll out their

portal to. It doesn't constrain you to only having SAP R/3 users as portal users, since non-SAP R/3 users can be defined in another store (e.g., LDAP, portal system database, etc.). Plus, if your SAP system is load balanced, you automatically gain a load-balanced data store!

- **A combination of an LDAP directory server and the portal system database:** When establishing a user authentication store using the portal administration console, you are able to specify an LDAP/database combination. This approach allows the portal administrator to add users to the portal without needing to contact the directory server administrator. It also allows additional attributes of a user to be stored in the database beyond what is already stored in the LDAP.

- **A combination of an SAP R/3 system and the portal system database:** This allows for the authentication of any users already present in the SAP R/3 system. All new users are created in the portal system database.

- **A text file:** This option is for testing only and should not be used in productive instances.

The user authentication store is one area where SAP EP 6.0 has significantly improved upon the SAP EP 5.0 offerings:

☑ **Read data from and write data to multiple repositories in parallel:** As of SAP EP 6.0, you are not tied to using just the LDAP as the data store. Portal users can now be stored in multiple locations.

☑ **Use different repositories for different sets of user attributes:** For instance, most user attributes can be pulled from SAP R/3, while other attributes such as user ID expiration date can be stored in the portal system database.

☑ **Store internal and external user data in separate directory servers:** You may also choose to have separate directory servers for different sets of portal users. For example, your internal users can be in your corporate LDAP, and the external portal users can reside in a new, dedicated directory server. This would allow for more security as well as let you specify different attributes for those users.

As you can see, SAP EP 6.0 offers more options and more flexibility to allow the portal to adjust to your existing user environment. More options should not mean more complexity, however. At the end of the day, adopt the easiest option for the task and needs at hand, whether it is the portal system database for quick configuration of a development system, an LDAP if one is in existence, the SAP R/3 system if it is compatible and includes the portal's users, or perhaps one of the available combinations.

How Will You Secure Network Communications?

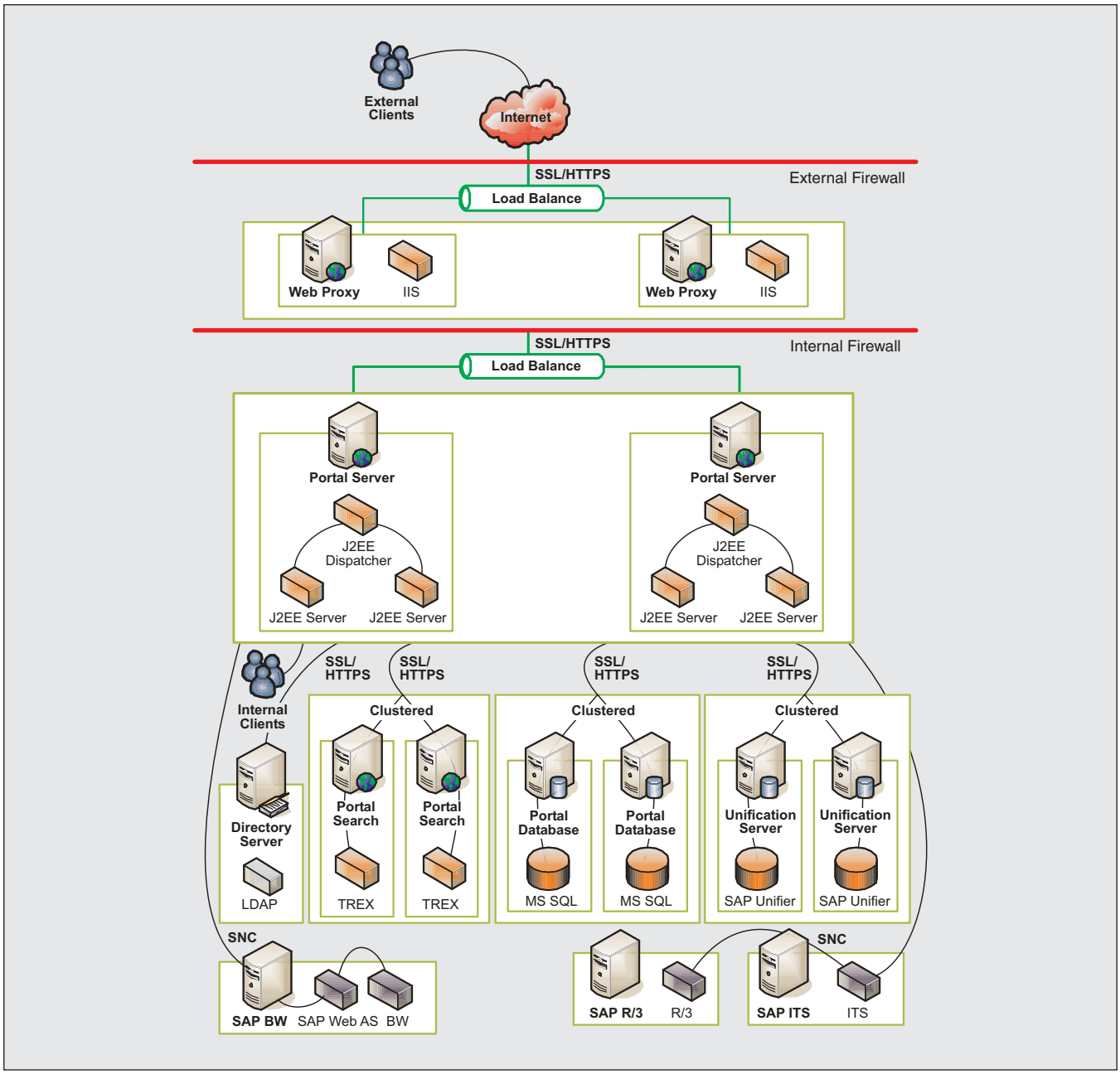
For communications security (see column ❺ in Figure 2), SAP EP supports two industry-standard technologies:

- **Secure Sockets Layer (SSL):** If you've purchased anything over the Internet, you've undoubtedly used SSL (you'll see *https* in the URL instead of *http*). With SSL, a certificate is installed on the portal server, and the browser and portal server then handle the rest, automatically negotiating cryptographic keys, and encrypting/decrypting and validating all communications.[13] You can generally obtain an SSL certificate for a few hundred dollars from a company like VeriSign. Note that it can take anywhere from one hour to two weeks to get the certificate (since the company needs to verify your identity), so plan accordingly.

- **Secure Network Communication (SNC):** SNC is a lesser-known protocol than SSL, but it has been used for years to encrypt communications

[13] In addition to protecting communications from unauthorized viewing through encryption, the browser, and Web servers, use a "check-sum" method to ensure that values have not been modified in transit.

Figure 8 *A "High-Security" Landscape*



between SAP clients and SAP systems for clients that use the DIAG protocol.[14]

[14] SAPGUI for Windows, SAPGUI for Java, and ITS all communicate with SAP systems like R/3 via the proprietary DIAG protocol (i.e., as opposed to an open protocol like HTTP), and must be secured using SNC. ITS's SAPGUI for HTML uses HTTP between the browser and ITS server, and DIAG between ITS and the SAP system. The former can be secured with SSL; the latter with SNC.

Figure 8 illustrates your options for communication security. In portal landscapes, SSL is used to secure HTTP communications (HTTPS), including browser-to-proxy, proxy-to-portal-server, portal-server-to-ITS, portal-server-to-LDAP, and portal-server-to-database communication. SNC is used to secure DIAG-based communications, including

SAPGUI-for-Windows-to-SAP-application-server and ITS-to-SAP-application-server communications.

Here are some important things to keep in mind when securing your portal:

- ☑ When securing browser-to-portal communication with SSL, make sure to SSL-enable either both or neither of the portal and Unification servers to avoid alerts in user browsers like "This page contains secure and insecure objects." Note that you must purchase and install separate certificates for the portal and Unification servers since they will generate different keys and have different domain names.

- ☑ You can selectively secure only a few resources that are highly sensitive and can greatly improve the performance of your portal by only SSL-enabling those resources. You can do this by simply installing a certificate on a backend system's Web server, which will automatically turn on its SSL capabilities.

- ☑ If browsers will need direct access to backend, Web-based systems (e.g., SAP BW or SAP R/3), make sure to SSL-enable these backend resources as well. If you don't, portal users will receive a message that says they are leaving a secure site when they access an iView based on that system. This can be frustrating and inconvenient.

When securing the communications between clients and backend systems, it is also essential to understand and focus on who is trying to talk to whom. Those new to SAP EP can find this very tricky for two reasons:

- SAP EP lets you launch native GUI applications on the frontend PC, like SAPGUI for Windows or SAPGUI for Java to access SAP transactions, IXOS-Viewer to view documents in an archive system, and Microsoft Excel to view Business Explorer (BEx) queries in SAP BW. Sometimes it's even hard to tell which GUI is being used, because the native GUI application is embedded within the browser window (in an iView).

- SAP EP offers a setting at the iView level called an "isolation method" (called an "isolation level" in SAP EP 5.0) that controls whether content is requested from the portal server or directly from the backend system. (For a description of the SAP EP 6.0 isolation methods, see the sidebar below.)

Understanding iView Isolation Methods

SAP EP includes a concept called "isolation methods" (previously called "isolation levels" in SAP EP 5.0) that controls the path content takes from the source system to the browser. In SAP EP 6.0, you can set an iView's isolation method by selecting "Load" from the property category menu in the iView editor. You'll find three options:

- **Embedded:** With this option, iView content is collected by the portal server and the finalized page is then presented to the browser. When the user clicks on any link or button the entire page is reloaded, which can be slow.

- **Isolated URL:** With this option, content is taken directly from the source to the browser. The benefit is that iView actions don't affect the rest of the page.

- **Pumped:** Like the embedded option, the portal server retrieves the content and downloads the entire page all at once. The difference is that the page is generated with an iFrame for each iView, and

(continued on next page)

(continued from previous page)

JavaScript code individually writes the content for each iView into its iFrame. The benefit of this option is that it avoids the page refreshing upon a user action (like isolated URLs), but still routes all traffic through the portal server so your users only need connectivity to the portal server.

From a landscape design perspective, the embedded and pumped options are nearly identical. In both cases, the portal server retrieves content from the backend system and delivers it to the browser. The isolated URL option, however, requires direct network connectivity between the browser and the backend system, something that you will specifically want to prevent for Internet users. Keep this in mind when defining iViews, and advise your support personnel to check if an iView is set to run as an isolated URL in case Internet users complain that a new iView won't load.

For more information, go to the SAP EP 6.0 SP3 documentation at **http://help.sap.com** and follow the path *Portal → Administration Guide → Content Administration → Portal Pages → Portal Page at Runtime → Isolation Method of iViews*.

Here are two rules to remember that will hopefully clarify things:

- ☑ If an iView uses a native GUI like SAPGUI for Windows or iXOS-Viewer, regardless of whether it is embedded within a browser window,[15] communications must be secured with SNC or some other means, not SSL. This is especially important to keep in mind when accessing the portal via the Internet, in part because you probably will not open the ports necessary for these native GUIs to function. The recommended approach is to limit use of these GUIs to internal use, sticking to content that can be transmitted via HTTP for Internet users not using VPN.

- ☑ If an iView contains Web-based content, it can be secured via SSL by installing an SSL certificate on the source system's Web server. Note that the SSL certificate is usually placed on the portal server, but beware if you intend to create iViews that use the isolated URL method in SAP EP 6.0 — the browser will directly request content from the backend system, and the SSL certificate must be installed on the backend system's Web server in order to avoid a "page not found" error.[16]

How Will You Implement Single Sign-On?

Single sign-on suppresses the logon dialog that normally appears when an iView accesses a backend system for the first time. While eliminating the need for repeated logons, and relieving users from having to remember multiple user IDs and passwords, it introduces greater risk — if a user leaves the workstation without logging out, the backend system is available to anyone who passes by. One way to secure sensitive applications is to write a custom ABAP "gatekeeper" transaction that asks for an SAP user ID and password before allowing access. It's also a good idea to set the portal sign-on to expire after a short period of inactivity.

[15] This is an option on the Load tab of certain iViews to have native GUIs launch directly on the desktop rather than embedded within a browser window.

[16] This is also true for the SAP EP 5.0 "by-pass" isolation mode, which tells the browser to bypass the portal server's iView rendering mechanism entirely and get the iView's content directly from the source system.

Once you've decided to implement single sign-on for one or more backend systems, you then have to decide how to accomplish it (see column ❻ in Figure 2). SAP EP offers two options: user mapping and logon tickets.

> ✓ ***Note!***
>
> *The type of authentication method you use has no effect on the options for single sign-on.*

With user mapping, users enter and maintain their list of user IDs and passwords for backend systems in the personalization area of their portal.[17] Thereafter, when an iView needs to access a backend system, the portal automatically passes the appropriate user ID and password to the backend system — the process is normally invisible to the user, and the user is only prompted if the user ID or password is invalid or has expired. This approach is inexpensive and is mostly used if portal user IDs are not identical to those in the backend systems.

Logon tickets offer a much better overall approach, but can only be used if the user IDs are consistent in both the portal's user authentication store and the SAP system. It involves installing a public key certificate, generated by the portal, on the backend system in order to establish a trust between the portal and backend system. When a user calls a backend application,[18] the logon ticket is passed to the application where it is checked for validity. It first checks to make sure that a trusted portal server has issued the ticket. Then the digital signature of the portal server is verified. Finally, if the ticket is deemed valid and trusted, the user ID is extracted and checked against the backend system's user registry. No password is sent since the backend system trusts that the user is who the portal says he or she is, and grants the user the access that he or she would normally have when logging in explicitly.

Even though user mapping is the easier single sign-on solution to implement, it is more complicated from the end user's perspective. Going to the portal's personalization page, selecting the appropriate backend system, and entering your user ID and password may seem like a simple, three-step process, but I have seen it cause significant confusion in certain client user communities. It may also give the impression that the portal is "broken" at first rollout, which can derail user acceptance of the portal. Administrators are also technically able to map the end users, but that task can be huge, even if they have the needed information. Single sign-on using a logon ticket, on the other hand, is the more graceful solution and should always be the first option in any implementation.

Helpful Hints

Here are some final tips to help you apply what you've learned here to your own environment and identified needs.

Sizing Your Production Portal Landscape

While we have discussed a large number of landscaping topics, you may be wondering how to size your portal boxes. SAP provides a Web-based sizing tool for SAP EP at **http://service.sap.com/quicksizing**, as well as for its other systems. A detailed discussion of each criteria is beyond the scope of this article, but here are a few considerations to keep in mind:

- ☑ The tool's output is highly dependent on its input. To obtain useful results, you need to diligently estimate several sizing inputs, including the number of concurrent portal users, the number of power users vs. casual browsers, the type of content rendered, the average number of roles displayed per user, and how many servers you'll

[17] The personalization area is accessed by clicking the "Personalize" hyperlink in the portal's global navigation area (i.e., the banner at the top of each portal screen).

[18] That is, accesses a page containing an iView that is either generated from or requires data from the backend system.

include in your landscape (e.g., for redundancy). There are also questions more specific to SAP Knowledge Management (SAP KM) repositories and an estimate of documents to be indexed. Obviously, these numbers can be hard to predict, but be generous with your estimates, as the portal is sure to grow with time. Also, additional CPU capacity and hard drive space is relatively inexpensive these days, and the cost of oversizing is always reasonable vs. the insurance it buys until you gain experience estimating your needs.

- ☑ Run the numbers based on an estimate of what you will need at six-month intervals over a period of two to three years (this was discussed in detail in my previous article). If your landscape has a load-balanced structure, you can always add servers as planned in your deliberate growth strategy.

- ☑ As a rule of thumb, a two-processor portal server with 2 GB of RAM can support up to 1,000 concurrent "normal-browsing" users.

- ☑ Some hardware vendors, like Dell and HP, have their own sizing methodology to guide you in sizing SAP EP servers. For more information, please visit their Web sites at **www.hp.com** and **www.dell.com**.

Building Your Development and Quality Assurance Landscapes

Although the technical options for development (DEV), quality assurance (QA), and production (PRD) environments are the same, DEV and QA environments face unique challenges. Here are some things to keep in mind.

Consider the Many Roles Each System Will Play

DEV environments are often used as a sandbox for developers to experiment and unit test in, for functional analysts to evaluate business packages, and for administrators to test new support packages or portal versions. QA environments are sometimes used for testing, sometimes for training, and occasionally as a backup for those without a high-availability production landscape. The latter can only be accomplished, however, if the QA system is designed to use the production user data store. Modeling a QA environment like the PRD environment, however, can be an expensive proposition, especially if the PRD environment is a highly redundant architecture involving multiple servers. When designing the DEV and QA landscapes, therefore, identifying the many roles of each system will be an important first step.

Consider Your Promote-to-Production Strategy

The DEV and QA landscape configurations should also take into account the strategy for promoting content to production. For example, when iViews in the DEV portal that are pointing to your DEV SAP R/3 system are transported to the QA portal, they should then point to the QA SAP R/3 system. In order to accomplish this easily, systems should be defined in each portal so that even though they are pointing to their respective backend systems, their aliases should be the same. For example, an iView in the DEV system pointing to the system "SAP_BW" would always point to the appropriate backend system after migration without any further intervention.

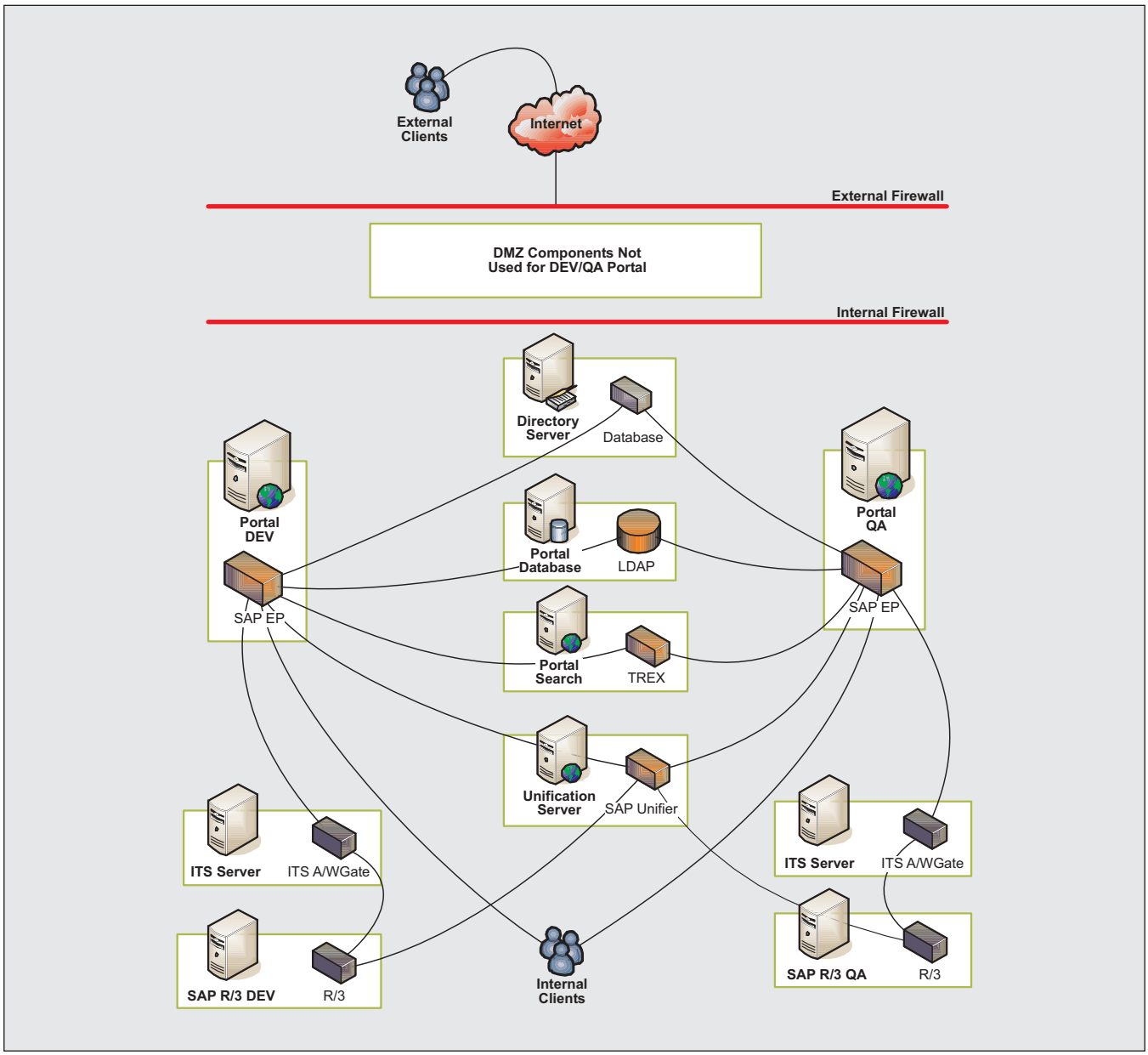
Consider Where You'll Evaluate Business Packages

SAP business packages generally come with a number of iViews ready for configuration, and it is rare that you will need all of them in your production system. Business packages are therefore generally installed in the DEV system, and only the appropriate iViews are configured and transported to QA and PRD. To keep your DEV system ultra-clean, consider importing business packages into a sandbox system for initial evaluation and only transporting iViews to DEV for final evaluation and configuration.

Consider Whether You Need Separate DEV and QA Systems, or If You Can Combine Them

DEV and QA systems can be combined in a single

Figure 9 Consolidated Development and Quality Assurance Environments



landscape, as shown in **Figure 9**, or they can have individual landscapes. There are criteria to consider and pros and cons for each approach. If the DEV landscape is completely independent of the QA landscape, it can more easily be used as a sandbox environment, as there would be no chance of affecting any shared QA resources. It would also have more leverage for installing new, unproven support packages and updates for all servers to test new functionality.

If the effort has been made to isolate the DEV environment, then the QA system should be made to more accurately reflect the PRD landscape. For example, if the PRD system is load balanced, then the QA system should be load balanced as well. If the PRD landscape utilizes the DMZ, then the QA landscape should do the same. By making the QA system resemble the PRD system as closely as possible, all testing done on it will be more realistic, and you will

have a better chance of identifying any potential problems before migration to PRD. This is a high-cost approach, though, involving more servers and more configurations. Cost can be controlled, however, by using smaller specification servers for QA, as this environment does not have to scale and supports a smaller number of users.

In my experience, the QA environment frequently borrows components from the DEV instance, and it makes economical sense to share certain elements. If server and implementation cost is of high importance, as it generally is, then you can make smart decisions and create a single landscape that caters to the requirements and functions of DEV and QA while keeping costs down by sharing certain components.

The first candidate for sharing between DEV and QA is the TREX server, if it is in scope. Since TREX requires its own server, it does not make much sense to have separate servers for each system. As a matter of fact, if further cost cutting is required, and if searching is not extremely critical, then the same TREX server can also potentially be used for PRD as well.

For similar reasons, another portal component that can be shared is the Unification server. The only caveat of sharing this or any other component is that new updates cannot be loaded onto the DEV system without affecting the QA testing environment.

The third candidate for server sharing is the database server. Multiple portal databases can be installed on the same database server. Unless PRD has its own server farm utilizing a storage array network (SAN), the same database server can also be used for PRD. If cost is extremely limited, then the database server can potentially be completely eliminated by installing the portal databases on the individual portal machines. This would eliminate the need for an additional database server or servers, but would increase the hardware-sizing requirements of the portal server.

Other candidates for sharing include components for Web-enabling backend applications such as ITS and SAP Web AS. These components are often installed on shared components. There could be separate installations of ITS and SAP Web AS for DEV and QA, but they could still share space on existing servers. For example, ITS can potentially be installed on the TREX or Unification server if available.

Conclusion

This two-part article series has illustrated the key requirements you need to consider to design and maintain a cost-effective portal landscape, and the key technical drivers of SAP Enterprise Portal landscapes. Landscaping your portal involves more than just sizing and setting up the hardware and software — these are the easy parts. Since the first phase of your portal implementation lays the groundwork for future growth (or limits it!), establishing a strategic landscape design early on is of utmost importance and should take up a significant portion of blueprint time. As a next step, I recommend you meet with your functional, business, and management teams, and use the issues and options herein as a basis for your initial design work. While agreeing on requirements will be the hardest part of your project, it will save your team a tremendous amount of time, money, and rework in the end.

Rizwan Uqaili is an Engagement Manager with Business Intelligence Services at Rapidigm. He also manages Enterprise Portal Services at Rapidigm, and has assisted numerous clients with their strategic SAP Business Intelligence and SAP Enterprise Portal initiatives and implementations. Rizwan has been a key speaker at ASUG conferences as well as the SAP BW and Portals conference. He graduated from the University of Texas at Austin and received his MBA from Clark University. Rizwan can be reached at ruqaili@rapidigm.com.

