Chapter 2

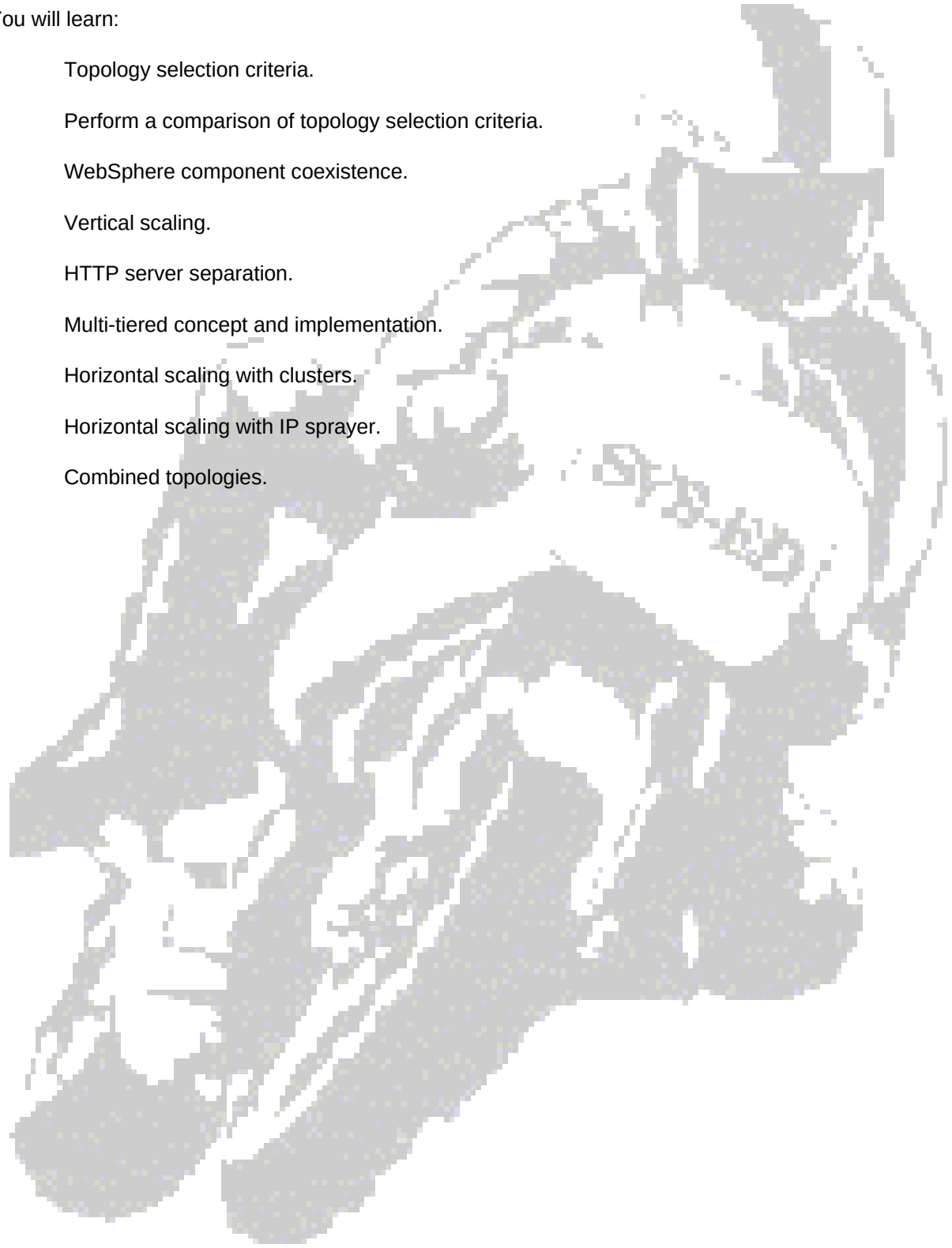
TOPOLOGY SELECTION

**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- Topology selection criteria.
- Perform a comparison of topology selection criteria.
- WebSphere component coexistence.
- Vertical scaling.
- HTTP server separation.
- Multi-tiered concept and implementation.
- Horizontal scaling with clusters.
- Horizontal scaling with IP sprayer.
- Combined topologies.



1 Topology Selection

There are a number of factors to consider when evaluating the topology for a WebSphere deployment.

The selection criteria include:

- Security
- Performance
- Throughput
- Scalability
- Availability
- Maintainability
- Session management

1.1 Security

Security concerns will require a physical separation of the web server from the application server processes.

A common configuration would be the use of two firewalls to create a demilitarized zone - DMZ. Information in the DMZ has protection based on protocol filtering between the Internet and the DMZ.

A web server intercepts the requests and forwards them on through the next firewall. The sensitive portions of the application and business data reside behind the second firewall, which filters based on IP addresses or domains.

1.2 Performance

Performance involves minimizing the response time for a given transaction load.

The two commonly used techniques are:

Vertical scaling	The creation of additional application server processes on a single physical machine, providing for software/application server failover as well as load balancing across multiple JVM application server processes. Vertical scaling allows an administrator to profile an existing application server for bottlenecks in performance, and use additional application servers, on the same machine.
Horizontal scaling	The creation of additional application server processes on multiple physical machines to take advantage of the processing power available on each machine. This will provide hardware failover support.

1.3 Throughput

Throughput involves the creation of a number of application server instances to service the same requests. WebSphere Application Server Network Deployment provides clusters, which logically group a number of application servers. The application servers are added through vertical and/or horizontal scaling

1.4 Scalability

Multiple machines can be configured to add processing power, improve security, maximize availability, and balance workloads.

The components that provide functions for configuring scalability include:

- WebSphere Application Server cluster support.

  Clusters allow the creation of a logical group of servers that all host and run the same application(s). Members of a cluster can be located on the same machine - vertical scaling and/or across multiple machines - horizontal scaling.

  The use of application server clusters can improve the performance of a server, simplify its administration, and enable the use of workload management.

- WLM: WebSphere Workload Management

  Incoming processing requests from clients are distributed among the clustered application servers. WLM enables both load balancing and failover, improving the reliability and scalability of WebSphere applications.

- IP sprayer

  Redirects incoming HTTP requests from web clients to a group of web servers. Although the clients behave as if they are communicating directly with a given web server, the IP sprayer will be intercepting all requests and distributing them among all the available web servers in the group.

  IP sprayers such as the Load Balancer component of Edge Components or Cisco Local Director can provide scalability, load balancing, and failover for web servers.

1.5 Availability

In order to avoid a single point of failure and maximize system availability, the topology must have some degree of process redundancy. High-availability topologies involve horizontal scaling across multiple machines.

Vertical scaling can improve availability by creating multiple processes, but the machine itself becomes a point of failure.

An IP sprayer will direct client HTTP requests to available Web servers, bypassing any that are offline. Another server can back up the IP sprayer, which eliminates it as a single point of failure.

A major benefit of scaling up to multiple machines will be improved scalability. WebSphere Application Server Network Deployment provides tools that can be used for managing availability by distributing critical functionality across multiple machines. Applications hosted on multiple machines will have less down time and are able to service client requests more consistently.

1.6 Maintainability

The topology affects the ease with which system hardware and software can be updated. The use of multiple WebSphere cells or horizontal scaling will make a system easier to maintain. Individual machines can be taken offline without interrupting other machines running the application.

It is important to recognize, that maintainability can conflict with other topology considerations. Limiting the number of application server instances makes the application easier to maintain; but will have a negative effect on throughput, availability, and performance.

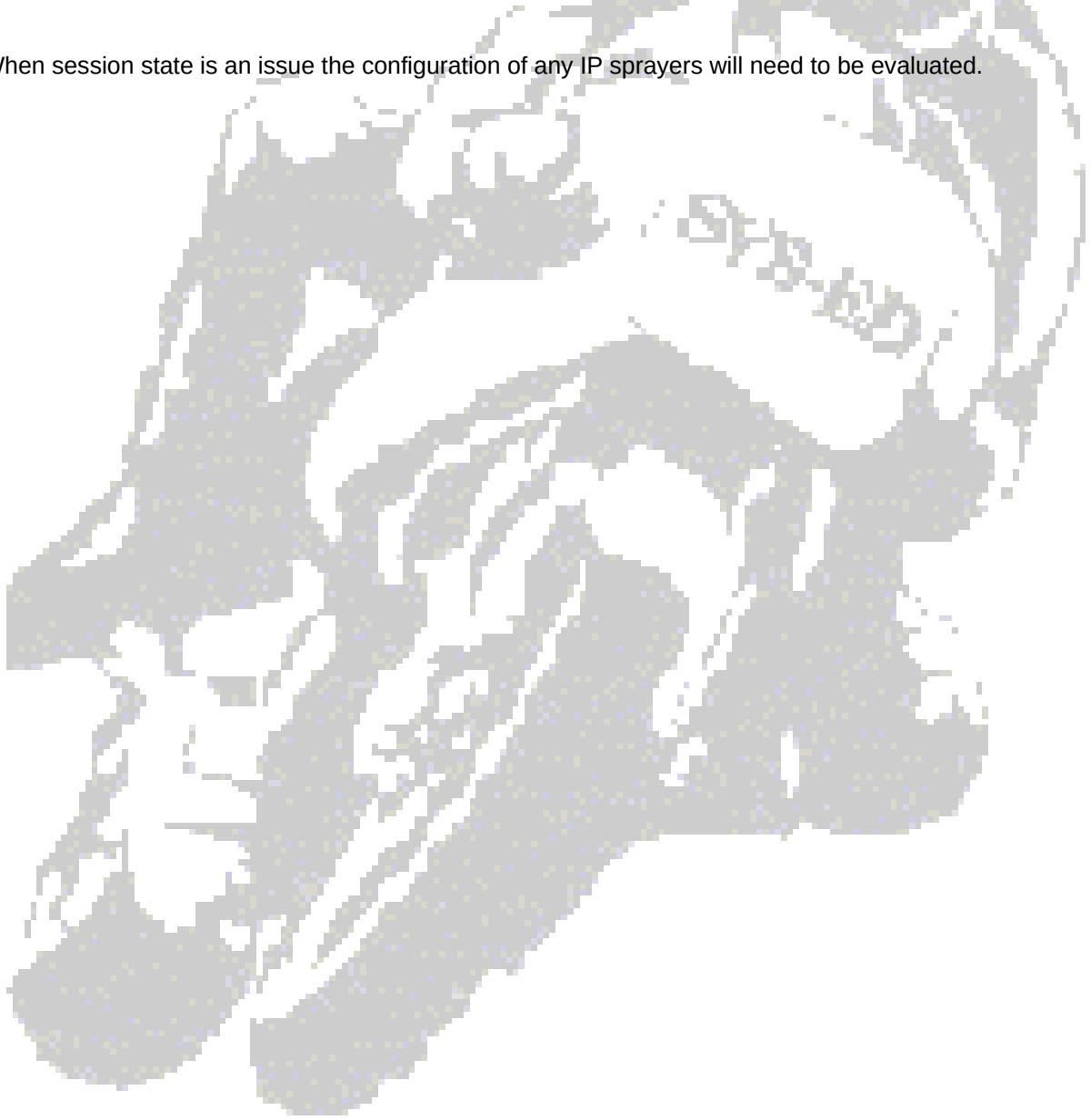
1.7 Session Management

Unless only a single application server is used or the application is completely stateless, maintaining session state between HTTP client requests will be a factor in determining the chosen topology.

Network Deployment provides two different functions for the sharing of sessions between multiple application server processes:

Database persistence	Session data is persisted to a database shared by the application servers.
Memory-to-memory replication	Provides replication of session data between the memories of different application server JVMs. WebSphere Internal Messaging is used to provide assured session replication between the JVM processes.

When session state is an issue the configuration of any IP sprayers will need to be evaluated.



2 Topology Selection Criteria

The following factors are not mutually exclusive and can be combined in a variety of different ways.

Topology	Security	Performance	Throughput	Maintainability	Availability	Session
Vertical scaling		Limited benefit	Limited to resources on a single machine	Easiest to maintain	Process isolation	Required
Horizontal scaling		Best in general	Best in general	Code migration to multiple nodes	Process and hardware redundancy	Required
HTTP separation	Allow for firewalls and DMZs	Usually better than local	Usually better than local			
Three tiers	Most options for firewalls	Typically slower than single JVM API	Clustering can improve throughput			
One cell				Ease of maintenance		
Multiple cells			Harder to maintain than single cell Process, hardware and software redundancy			

3 WebSphere Component Coexistence

WebSphere Application Server Network Deployment supports a number of scenarios for the coexistence of installations on a single machine.

There are new options to consider when selecting a topology:

- Multiple WebSphere Application Server instances.
- Multiple server instances using a single installation.
- Coexistence of WebSphere Application Server and Network Deployment.
- Single versus multiple web servers.

3.1 Multiple WebSphere Instances

WebSphere Application Server supports two types of runtime instances on a single machine:

Application server	Multiple instances from a single installation of WebSphere Application Server.
Deployment Manager	Multiple instances from a single installation of WebSphere Application Server Network Deployment.

Having multiple installations of specific software on a single machine is currently not supported.

However, a single installation can be configured to run multiple runtime instances on a single machine.

Installed product	**With WebSphere Application Server V5.0 base**	**With WebSphere Application Server V5.0 ND**
WebSphere Application Server V5 base	Supported Port conflicts can be resolved at installation time.	Supported There are no issues. There are no port conflicts in the default configuration settings for both products. The WebSphere development community commonly uses this environment.
WebSphere Application Server Network Deployment	Supported	Not supported at the current time.

3.2 Multiple Server Instances Using a Single Installation

The following configurations are supported for running multiple runtime instances using a single installation:

- Creating and running multiple WebSphere Application Server instances from a single WebSphere Application Server installation.
- Creating and running multiple Deployment Manager instances from a single WebSphere Application Server Network Deployment installation.

Multiple Deployment Managers can be configured for a single Network Deployment installation, however, they do not have the capability to provide failover or clustering support for each other. This configuration will allow the Deployment Managers of multiple, unconnected cells to be configured and run on a single machine.

3.3 Coexistence of WebSphere Application Server and Network Deployment

Network Deployment software can be installed on the same machine as WebSphere Application Server.

The advantages are:

- There is no need for a dedicated machine to host the cell Deployment Manager and its master cell repository.
- The ability to reuse existing backup facilities provided for the node machine.

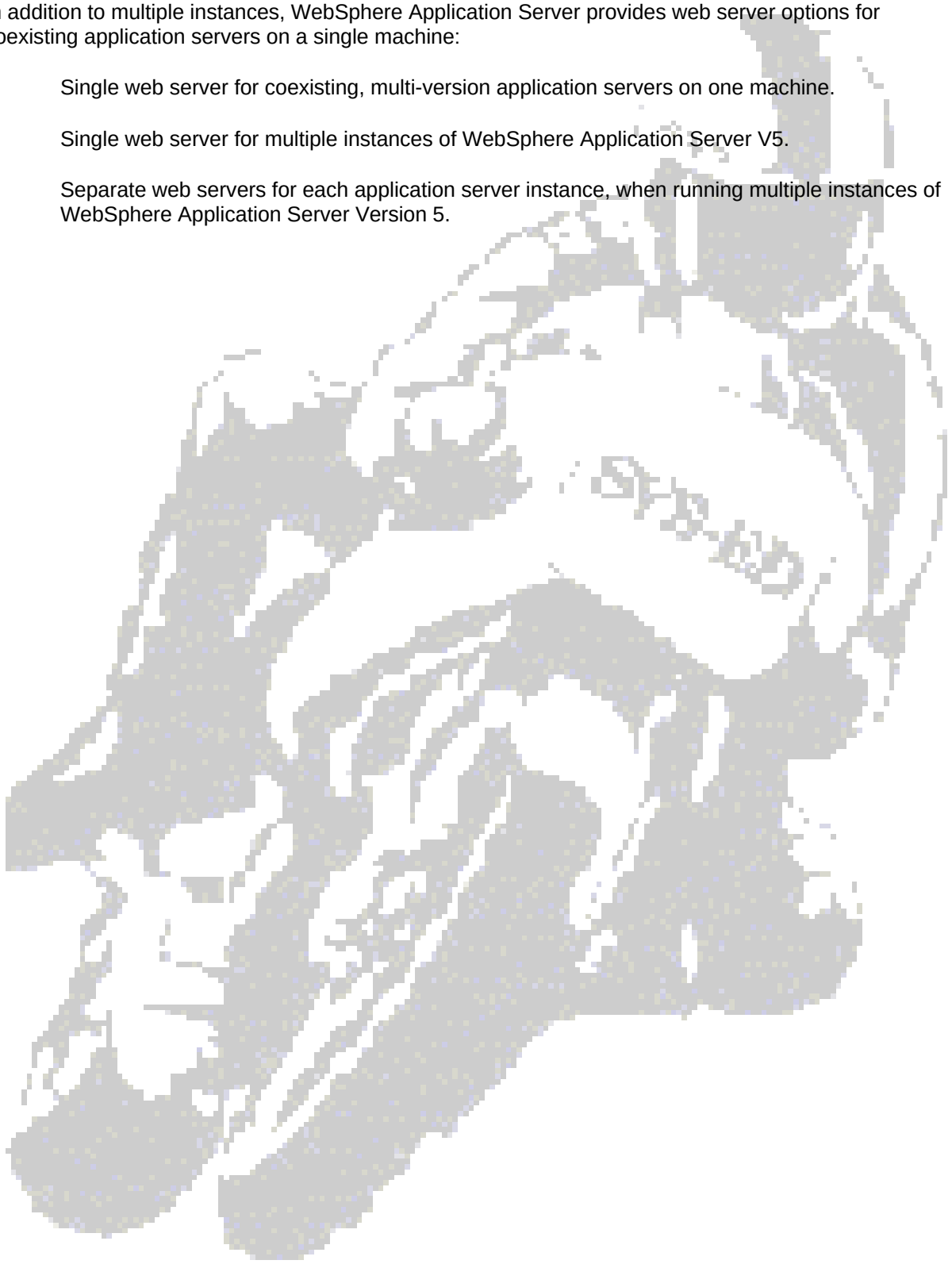
If something happens to either the WebSphere Application Server or Deployment Manager installation, then this will require that the nodes be rebuilt. Also the other component would need to be moved.

Having both the network deployment software and server would cause contention for system resources.

3.4 Single versus Multiple Web Servers

In addition to multiple instances, WebSphere Application Server provides web server options for coexisting application servers on a single machine:

- Single web server for coexisting, multi-version application servers on one machine.
- Single web server for multiple instances of WebSphere Application Server V5.
- Separate web servers for each application server instance, when running multiple instances of WebSphere Application Server Version 5.



4 Topologies

WebSphere Application Server Network Deployment supports a number of common topologies:

- Vertical scaling
- HTTP server separation
- Reverse proxy
- Multi-tiered
- Horizontal scaling with clusters
- Horizontal scaling with IP sprayer
- Multiple WebSphere cells
- Multiple clusters on a node
- Combined

4.1 Web Application Server Node

A web application server node is an application server that includes an HTTP server and is typically designed for access by HTTP clients and to host both presentation and business logic.

The web application server node is a functional extension of the informational publishing-based web server. It provides the technology platform and contains the components to support access to both public and user-specific information by users employing web browser technology.

The node can contain these data types:

- HTML text pages, images, multimedia content to be downloaded to the client browser
- Servlets, JavaServer Pages
- Enterprise beans

Application program libraries, such as Java applets for dynamic download to client workstations. In these topologies the web application server node is implemented using one or both of the following:

- WebSphere Application Server and the embedded HTTP transport.
- WebSphere Application Server, a supported web server, and the web server plug-in.

4.2 Web Server Redirector Node and Application Server Node

Most topologies will have a stand-alone web server residing in a DMZ between two firewalls.

In order to separate the web server function from the web application server function, the web application server must be split into two new nodes:

• web server redirector	• application server node.

Web clients send requests to the web server redirector node, which serves the static content such as HTTP pages.

Requests for dynamic content, requiring processing by servlets, JSPs, enterprise beans, and back-end applications are forwarded to the application server.

In these topologies, the web server redirector is implemented using a supported web server and the appropriate web server plug-in. The application server node is implemented using WebSphere Application Server.

4.3 Domain and Protocol Firewall Nodes

A firewall is a hardware/software system that manages the flow of information between the Internet and an organization's private network. A firewall can separate two or more parts of a local network to control data exchange between departments.

Components of firewalls include filters or screens, each of which controls transmission of certain classes of traffic. Firewalls provide the first line of defense for protecting private information. Security systems combine firewalls with encryption and other complementary services, such as content filtering and intrusion detection.

Firewalls control access from a less trusted network to a more trusted network.

Traditional implementations of firewall services include:

- Screening routers; which is the protocol firewall.
- Application gateways; which is the domain firewall.

A pair of firewall nodes provide levels of protection at the expense of increasing computing resource requirements. The protocol firewall is typically implemented as an IP router.

4.4 Directory and Security Services Node

The directory and security services node supplies information on the location, capabilities, and attributes of resources and users known to this web application system. This node can supply information for security services such as authentication and authorization and can also perform the actual security processing.

The authentication in most designs validates the access to the web application server part of the web server; the node will also authenticate for access to the database server.

4.5 Web Presentation Server Node

The web presentation server node provides services to enable a unified user interface. It is responsible for all presentation-related activity. In its simplest form, it serves HTML pages and runs servlets and JSPs. For more advanced patterns, it acts as a portal and provides the access integration services.

The web presentation server node interacts with the personalization server node to customize the presentation based on the individual user preferences or on the user role. The web presentation server allows organizations and their users to standardize and configure the presentation of applications and data in the most efficient way, while enabling fine-grained access control.

In these topologies, the Web presentation server is implemented using WebSphere Application Server. Function is basically limited to that which happens in the Web container. Enterprise beans reside on an application server node.

4.6 Database Server Node

The database server node provides a persistent data storage and retrieval in support of the user-to-business transactional interaction. The data stored is relevant to the specific business interaction.

This mode of database access will be important in determining the performance of this web application.

The recommended approach is to collapse the database accesses into single or very few calls. One approach for achieving this is by coding and invoking stored procedure calls on the database.

4.7 Deployment Manager

In a Network Deployment environment, the Deployment Manager is the focal point for configuration and operation. For each of the topologies, it will be necessary to make a decision regarding the placement of the Deployment Manager and master cell repository.

The Deployment Manager can be located on a dedicated machine or it can co-exist with a WebSphere Application Server installation. The general recommendation is to place it on a dedicated machine.

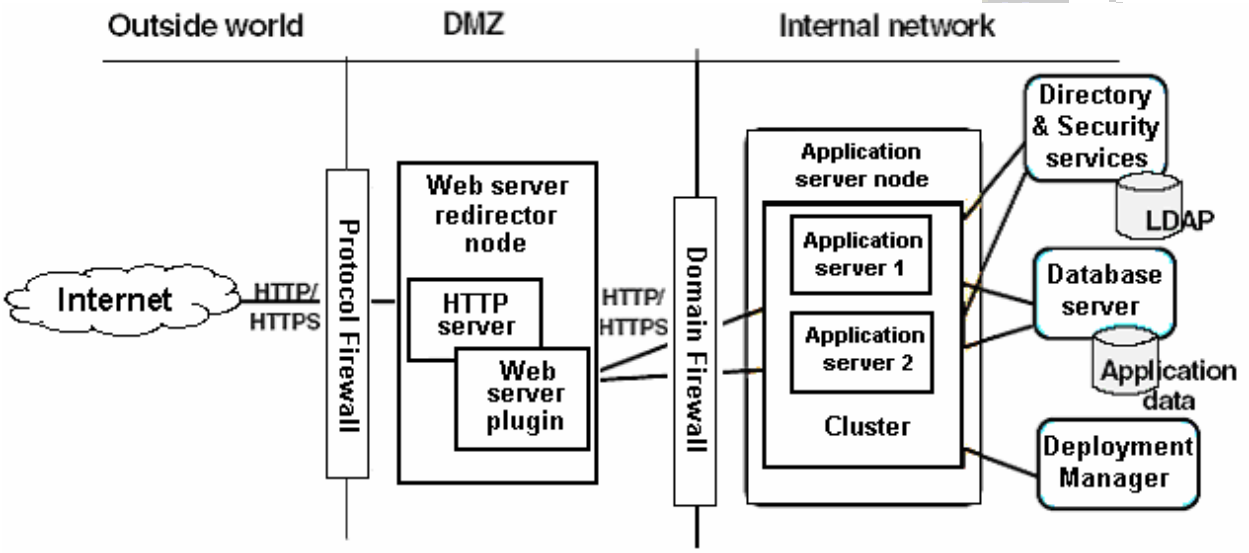
4.8 Load Balance Node

The load balancer node provides horizontal scalability by dispatching HTTP requests among several identically configured Web server or Web server redirector nodes. The use of a load balancer will also have an impact on the decision to implement session affinity.

The load balancer node is implemented using the Edge Components.

5 Vertical Scaling

Vertical scaling refers to configuring multiple application servers on a single machine,. This is commonly done by creating a cluster of associated application servers all hosting the same application(s).



This figure represents a simple vertical scaling example, with multiple cluster members on the application server node.

Vertical scaling can also be implemented multiple machines in a configuration.

Vertical scaling can be combined with other topologies to boost performance and throughput.

5.1 Advantages

Vertical scaling has the following advantages:

- Efficient use of machine processing power.

  An instance of an application server runs in a single JVM: Java virtual machine process. The inherent concurrency limitations of a JVM process prevent it from fully utilizing the processing power of a machine.

  Creating additional JVM processes provides multiple thread pools, each corresponding to the JVM associated with each application server process. This avoids concurrency limitations and permits the application server to use the full processing power of the machine.

- Load balancing.

  Vertical scaling topologies can make use of the WebSphere workload management.

- Process failover.

  Vertical scaling can provide failover support among application servers of a cluster. If one application server instance goes offline, the other instances on the machine continue to process client requests.

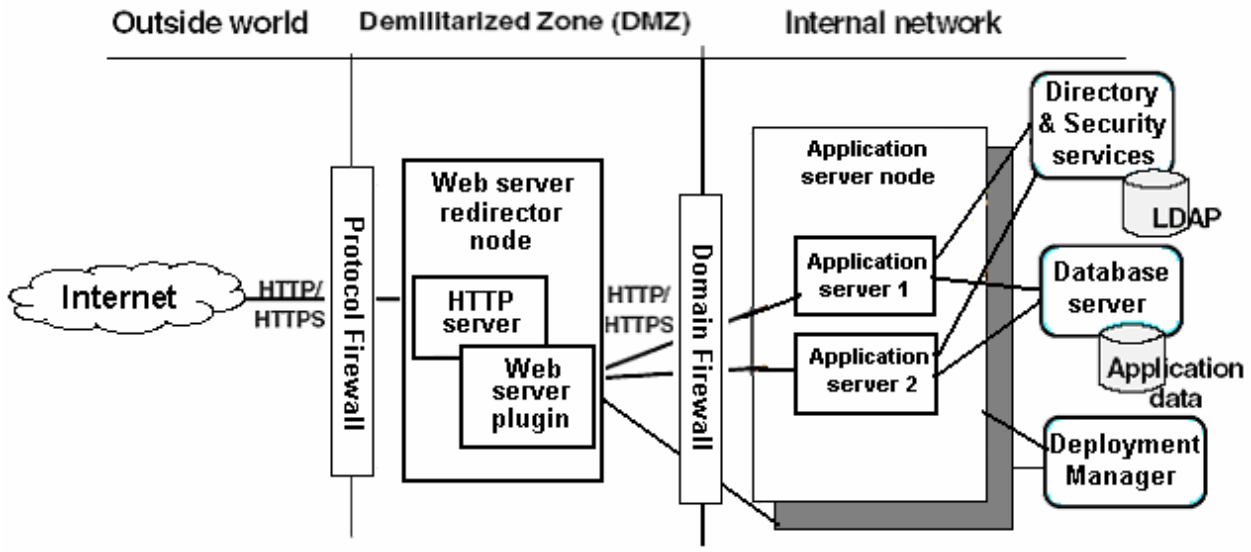
5.2 Disadvantages

Single machine vertical scaling topologies have the drawback of introducing the host machine as a single point of failure in the system.

6 HTTP Server Separation

HTTP server separation topologies physically separate the HTTP server from the application servers; this typically is achieved by placing the HTTP server in a DMZ. The use of a DMZ provides an additional layer of security for back-end servers and data.

WebSphere Application Server provides a web server plug-in for use on the web server machine. 'The Web server plug-in routes requests to application servers on remote machines using the HTTP or HTTPS protocol.



6.1 Advantages

HTTP server separation has the following advantages:

- Supports load balancing and failover, eliminating single points of failure.

  A point of failure exists when one process or machine depends on another process or machine. If the point fails, the whole system will become unavailable. When comparing DMZ solutions, a single point of failure refers to a single point of failure between the Web server and application server.

  Failover configurations can minimize downtime and possibly even prevent a failure.

- Avoids data access from DMZ.

  A DMZ configuration protects application logic and data by creating a demilitarized zone between the public web site and the servers and databases where the information is stored. It is important that DMZ topologies do not have servers that directly access databases from the DMZ.

- Supports NAT: Network Address Translation firewalls.

  A firewall product that runs NAT receives packets for one IP address, and translates the headers of the packet to send the packet to a second IP address.

  Environments with firewalls employing NAT, need to avoid configurations involving complex protocols in which IP addresses are embedded in the body of the IP packet - Java Remote Method Invocation (RMI) or Internet Inter-Orb Protocol (IIOP).

- Supports SSL: Secure Sockets Layer encryption for communications between the web server and the application server.

  Configurations that support encryption of communication between the web server and application server reduce the risk that attackers will be able to obtain secure information.

- Performance bottlenecks may be reduced.

- Administration is simplified. The Web server plug-in uses a single, easy-to-read XML configuration file.

6.2 Disadvantages

HTTP server separation has the following disadvantages:

- The link between the Web server and WebSphere Application Server is done using a web server plug-in.

  The plug-in will need to be configured after certain configuration changes from the WebSphere Application Server and manually moved to the proper location on the Web server.

- There is no protocol shift for inbound and outbound traffic across a firewall.

  The web server sends HTTP requests to application servers behind firewalls and an HTTP port in the firewall must be open to let the requests through.

  Configurations that require switching to another protocol and opening firewall ports corresponding to the protocol, are often more complex to set up, and the protocol switching overhead can impact performance.

  The following approaches will address this issue:

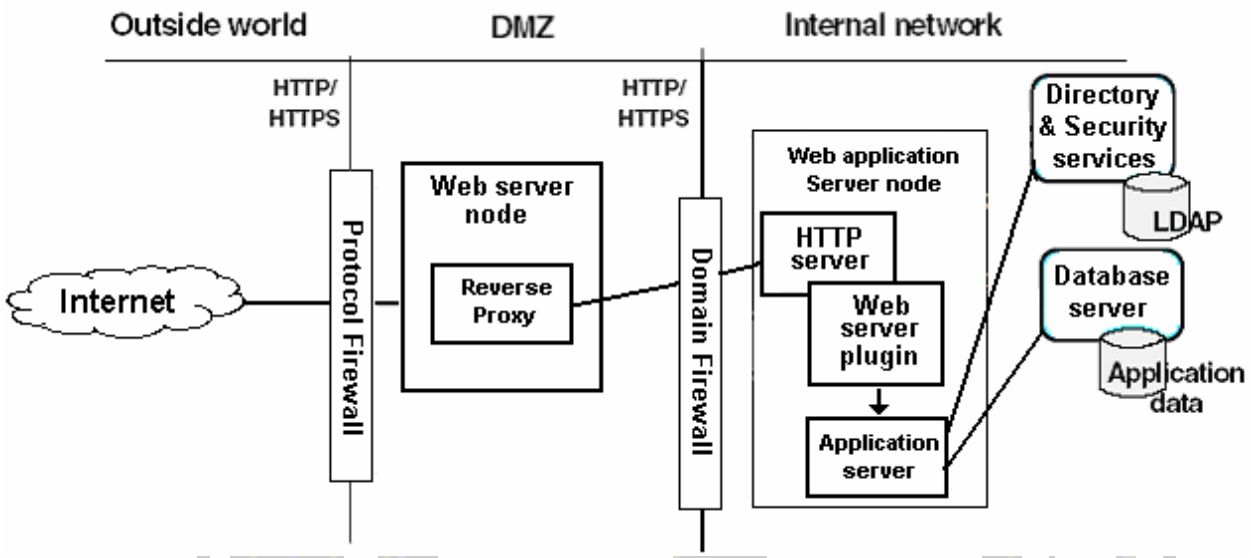
  1. Configure the web server plug-in to use HTTPS.
     - This will provide a high-security connection between the HTTP server and the application server.
     - This connection can be configured so that the Web server plug-in and application server must mutually authenticate each other using PKI: public-key infrastructure.
  2. Use different inbound browser to HTTP server and outbound web server plug-in to application server port numbers.

Characteristic	**Comment**
SSL support	Yes
Workload management	Yes
Network Address Translation (NAT)	Yes
Performance	High
Administration of configuration	Manual
Avoids data access from DMZ	Yes
Avoids DMZ protocol switch	No
Avoids single point of failure	Yes
Compatible with WebSphere security	Yes
Minimum required number of firewalls holes	1 per application server, plus 1 if WebSphere Application Server V5 security is used by the Web server.

6.3 Reverse Proxy

Reverse proxy or IP-forwarding topologies use a reverse proxy server to receive incoming HTTP requests and forward them to a Web server. The web server in turn forwards the requests to the application servers that do the actual processing.

The reverse proxy returns requests to the client, hiding the originating web server.



Reverse proxy servers are typically used in DMZ configurations to provide additional security between the public Internet and the Web servers and application servers servicing requests. A reverse proxy product used with WebSphere Application Server must support NAT: Network Address Translation.

Reverse proxy configurations support high-performance DMZ solutions that require as few open ports in the firewall as possible. The reverse proxy capabilities of the web server inside the DMZ require as few as one open port in the second firewall.

6.4 Advantages

Advantages of using a reverse proxy server in a DMZ configuration include:

- That is a well-known and tested configuration, resulting in less customer confusion than other DMZ configurations.
- It is a reliable and fast-performing solution.
- It eliminates protocol switching by using the HTTP protocol for all forwarded requests.
- It has no effect on the configuration and maintenance of a WebSphere application.

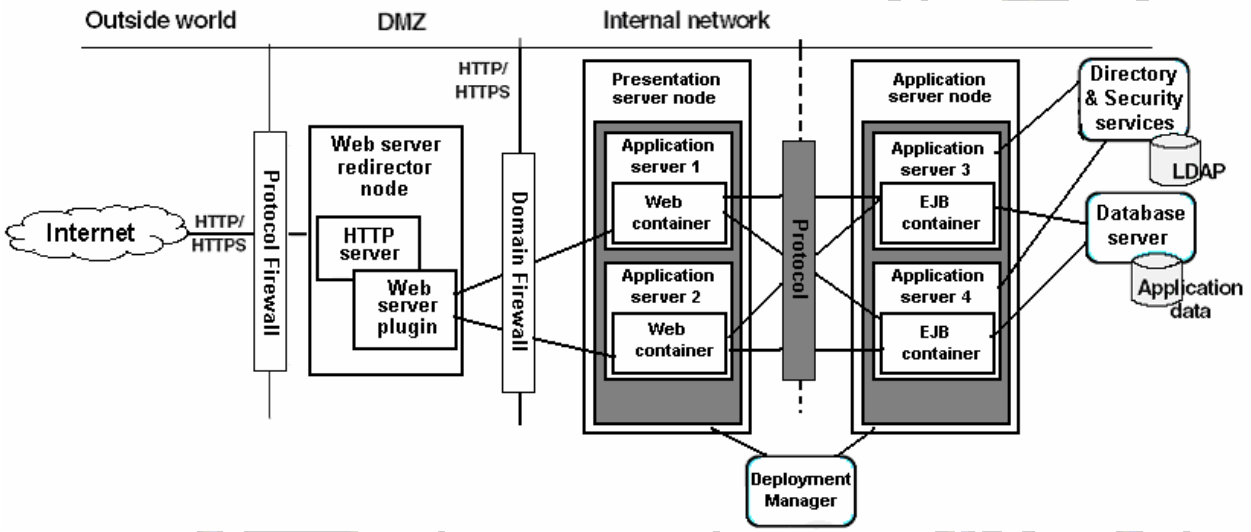
6.5 Disadvantages

Disadvantages of using a reverse proxy server in a DMZ configuration include:

- It requires more hardware and software than similar topologies that do not include a reverse proxy server; therefore it will be more complicated to configure and maintain.
- That the reverse proxy does not participate in WebSphere workload management.
- It can't be used in environments where security policies prohibit the same port or protocol being used for inbound and outbound traffic across a firewall.

7 Multi-tiered

Partitioning the application server processes into servlet application servers and EJB application servers can provide advantages from a security perspective, as well as some potential advantages from a performance perspective.



Application server processes that run servlets reside on a front-end node; this is referred to as a presentation server node. This will put the servlet execution closer in a network sense to the HTTP server; thereby, improving response time to client requests.

Application server processes that run enterprise beans reside on the application server node. This puts them in closer proximity to the application data, which is represented in an application by entity beans and stored on the database server. Clustering the application servers will help to maximize resource use on each node. It will provide process redundancy and use memory more efficiently than in similar topologies that host only single instances of application servers.

7.1 Advantages

Multi-tiered topologies have the following advantages:

- Allows flexibility in the replication of application servers..
- Additional resources on the machines can improve application throughput and performance.
- If a firewall is introduced between each pair of tiers, an additional layer of security can be provided for entity beans and application data.

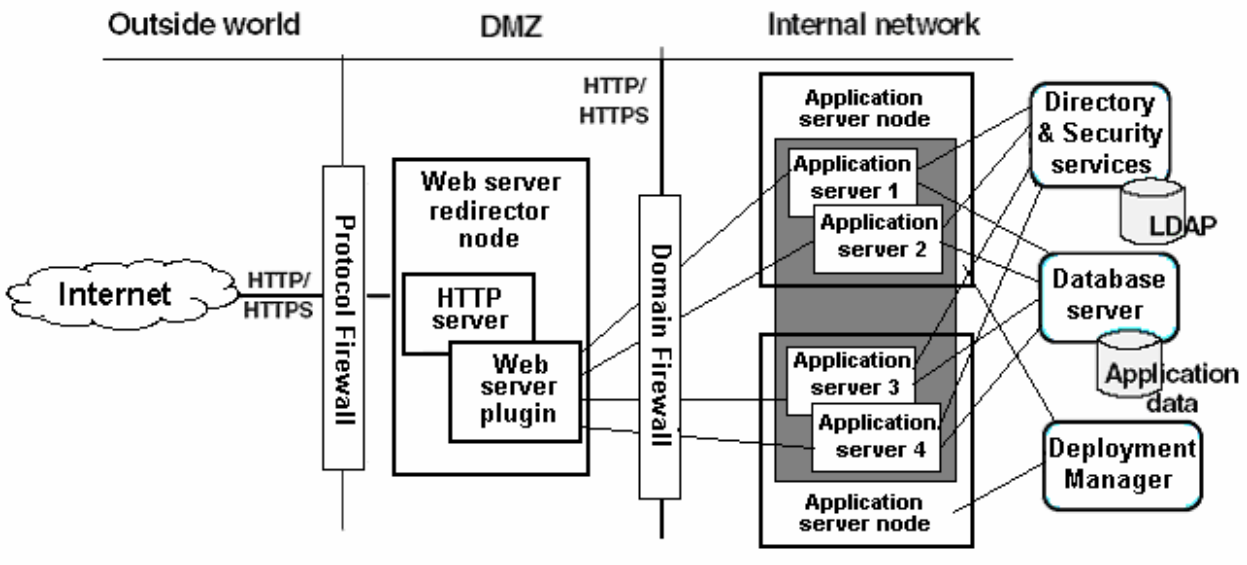
7.2 Disadvantages

Multi-tiered topologies have the following disadvantages:

- Eliminates the local JVM: Java virtual machine optimizations that occur when both the web containers and EJB containers run in the same application server. It also introduces network latency. These factors tend to slow system performance.
- The level of redundancy can make maintenance more complicated.

8 Horizontal Scaling with Clusters

Horizontal scaling exists when the members of an application server cluster are located across multiple physical machines. This lets a single application span several machines, yet still present a single logical image.



The web server plug-in distributes requests to cluster member application servers on the application server nodes.

The Network Dispatcher component of Edge Components, which has the capability for distributing client HTTP requests, can be combined with clustering to reap the benefits of both types of horizontal scaling.

8.1 Advantages

Horizontal scaling using clusters has the following advantages:

- Provides the increased throughput of vertical scaling topologies; also provides failover support. This topology allows handling of application server process failure and hardware failure without significant interruption to client service.
- Optimizes the distribution of client requests through mechanisms such as workload management or remote HTTP transport.

8.2 Disadvantages

Horizontal scaling using clusters has the following disadvantage:

- Increased maintenance effort.

9 Horizontal Scaling with IP Sprayer

Load-balancing products can be used to distribute HTTP requests among application server instances that are running on multiple physical machines.

The Network Dispatcher component of Edge Components is an IP sprayer that performs load balancing among web servers.

9.1 Simple IP Sprayer Topology

A simple horizontal scaling configuration will use an IP sprayer on the load balancer node to distribute requests among application servers on multiple machines:

A backup node for the load balancer node is normally configured in order to eliminate it as a single point of failure.

9.2 Complex IP Sprayer Topology

A complex IP sprayer topology will use an IP sprayer for distributing requests among several machines containing Web servers and clustered application servers.

In an IP sprayer or similarly configured topology, each web server can be configured to perform workload management over all application servers in a specific cluster. In this configuration, session affinity will be preserved, regardless as to which web server is passed the request by the IP sprayer.

9.3 Minimal Clustered

In a minimal clustered scenario, the business logic tier application servers would be clustered and host the servers hosting the EJBs. The application servers in this tier will act as a common service layer to the application servers in the presentation tier. As a cluster, they would be a single logical application server.

The application servers in the presentation tier would be non-clustered. Each would receive a request from its local web server via the web server plug-in, process the request in its JVM on that machine, and then call the logical business logic layer to take advantage of EJS workload management.

9.4 Complex Clustered

Complex clustered extends minimal clustered by clustering the presentation tier application servers.

The web server plug-in on each physical server can be configured to distribute requests across all the presentation application servers. In this way if the local application server dies, the web server plug-in can perform its normal failover support to push the request to one of its other configured presentation application servers.

In a situation when the web server process on a particular machine dies, the presentation application server JVM will still receive requests from the web servers on the other machines. Likewise, the business logic application servers would still be clustered to take advantage of EJS workload management.

9.5 Advantages

The advantages in using an IP sprayer to distribute HTTP requests are:

- Improved server performance by distributing incoming TCP/IP requests among a group of web server machines.
- The use of multiple web servers increases the number of connected users.
- Elimination of the web server as a single point of failure. It can also be used in combination with WebSphere workload management to eliminate the application server as a single point of failure.
- Improved throughput by letting multiple servers and CPUs handle the client workload.

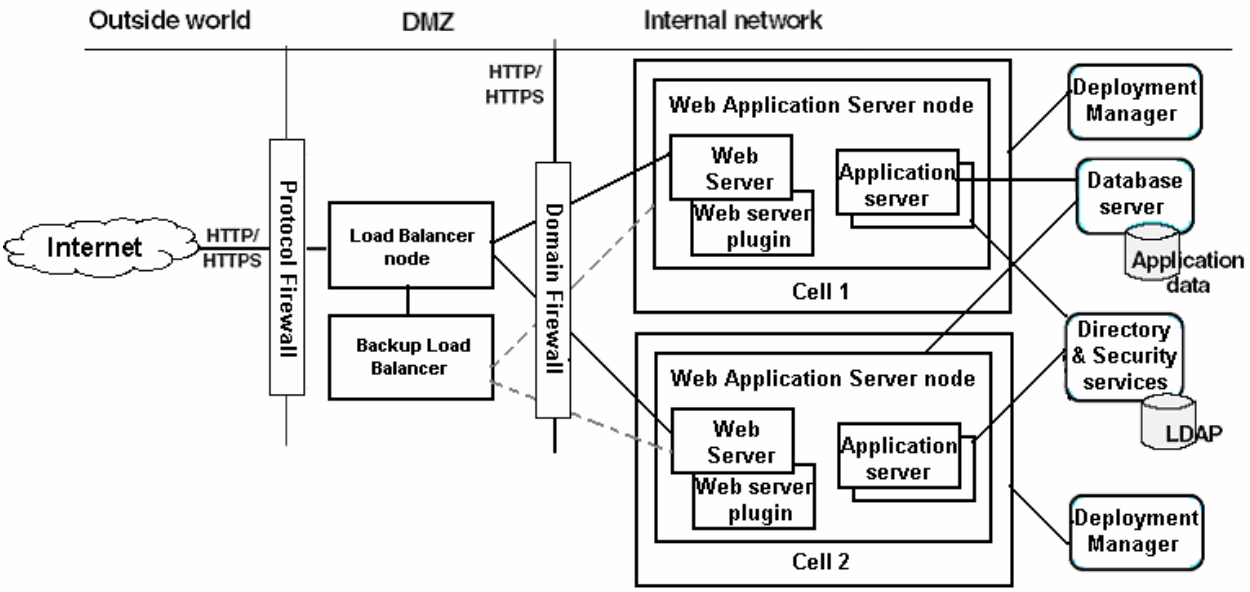
9.6 Disadvantages

The disadvantage of using of an IP sprayer to distribute HTTP requests is:

- Extra hardware and software are required for the IP sprayer servers.

10 Multiple WebSphere Cells

The application in this diagram runs simultaneously in two cells, each hosted on a different physical machine. The load balancer node is used as an IP sprayer to distribute incoming HTTP requests among the two cells, presenting a single image of the application to clients.



Each has its own set of XML configuration files and is administered independently. A different version of the application can be run in each cell. Since the cells are isolated from one another, different versions of the WebSphere Application Server software can be used in each cell.

There are no hard limits on the number of nodes and application servers that can be used in a single cell.

10.1 Advantages

The advantages in using multiple cells are:

- Isolation of hardware failure

  If one cell goes offline due to hardware problems, the others can still process client requests.

- Isolation of software failure

  Running an application in two or more cells isolates any problems that occur within a cell, while the other cells continue to handle client requests.

  This can be helpful when:

  - Rolling out a new application or a revision of an existing application. The new application or revision can be brought online in one cell and tested in a live situation while the other cells continue to handle client requests with the production version of the application.
  - Deploying a new version of the WebSphere Application Server software. The new version can be brought into production and tested in a live situation without interrupting service.
  - Applying fixes or patches to the WebSphere Application Server software. Each cell can be taken offline and upgraded without interrupting the application.

- If an unforeseen problem occurs with new software, using multiple cells can prevent an outage to an entire site.

  A rollback to a previous software version can also be accomplished more quickly. Hardware and software upgrades can be handled on a cell-by-cell basis during off-peak hours.

- Improved performance

  Running an application using multiple smaller cells may provide better performance than a single large cell because there is less inter-process communication in a smaller cell.

10.2 Disadvantages

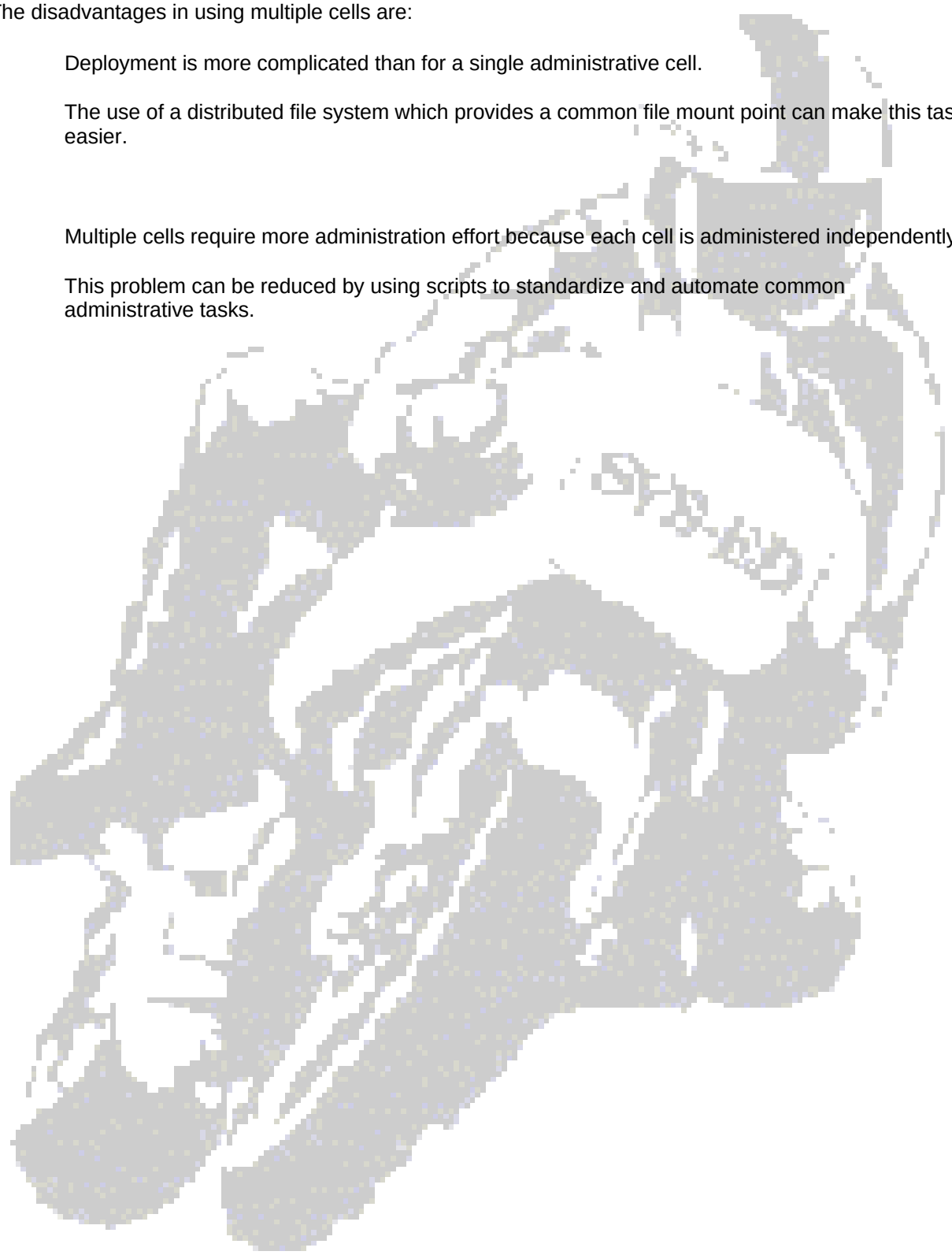
The disadvantages in using multiple cells are:

- Deployment is more complicated than for a single administrative cell.

  The use of a distributed file system which provides a common file mount point can make this task easier.

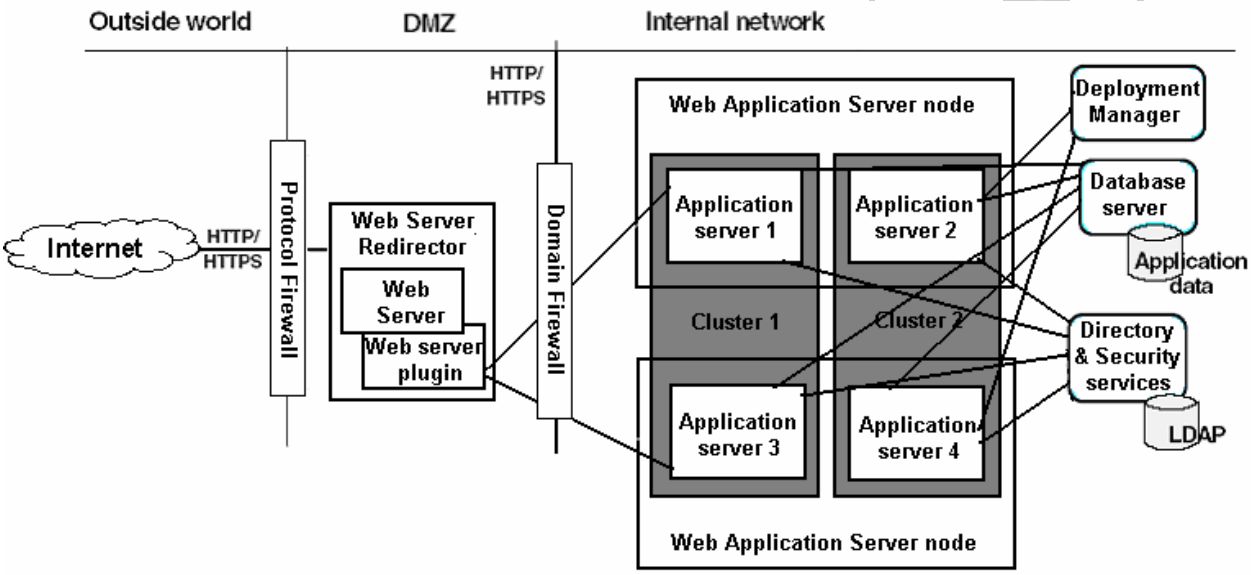
- Multiple cells require more administration effort because each cell is administered independently.

  This problem can be reduced by using scripts to standardize and automate common administrative tasks.



11 Multiple Clusters on a Node

When deploying an application, a decision will have to be made as to whether to deploy a cluster of application servers across all machines.



Each cluster of application servers is distributed throughout all of the machines in the system. In this example, a member of each cluster is hosted on each Web application server Variation: Another possible multi-cell configuration exists, although it's more limited. The determining factor is that both WebSphere cells must be running the same version of WebSphere. When this is the case, you can configure a Web server and its Web server plug-in to service multiple cells simultaneously.

The configuration consists of:

- A Web server and Web server plug-in on a machine in the DMZ.
- Multiple cells and application servers configured on (and across) multiple machines behind the domain firewall.

Manual manipulation of the Web server plug-in configuration file in each cell would be required so requests are sent to application servers residing in multiple cells. Session affinity between application servers in the same cell would work, but not between application servers in multiple cells.

11.1 Advantages

Hosting members of multiple application server clusters across one or more machines has the following advantages:

- Improved throughput

  The use of an application server cluster enables the handling of more client requests simultaneously.

- Improved performance

  Hosting cluster members on multiple machines enables each member to make use of the machine's processing resources.

- Hardware failover

  Hosting cluster members on multiple nodes isolates hardware failures and provides failover support. Client requests can be redirected to the application server members on other nodes if a node goes offline.

- Application software failover

  Hosting cluster members on multiple nodes also isolates application software failures and provides failover support if an application server stops running. Client requests can be redirected to cluster members on other nodes.

- Process isolation

  If one application server process fails, the cluster members on the other nodes are unaffected.

11.2 Disadvantages

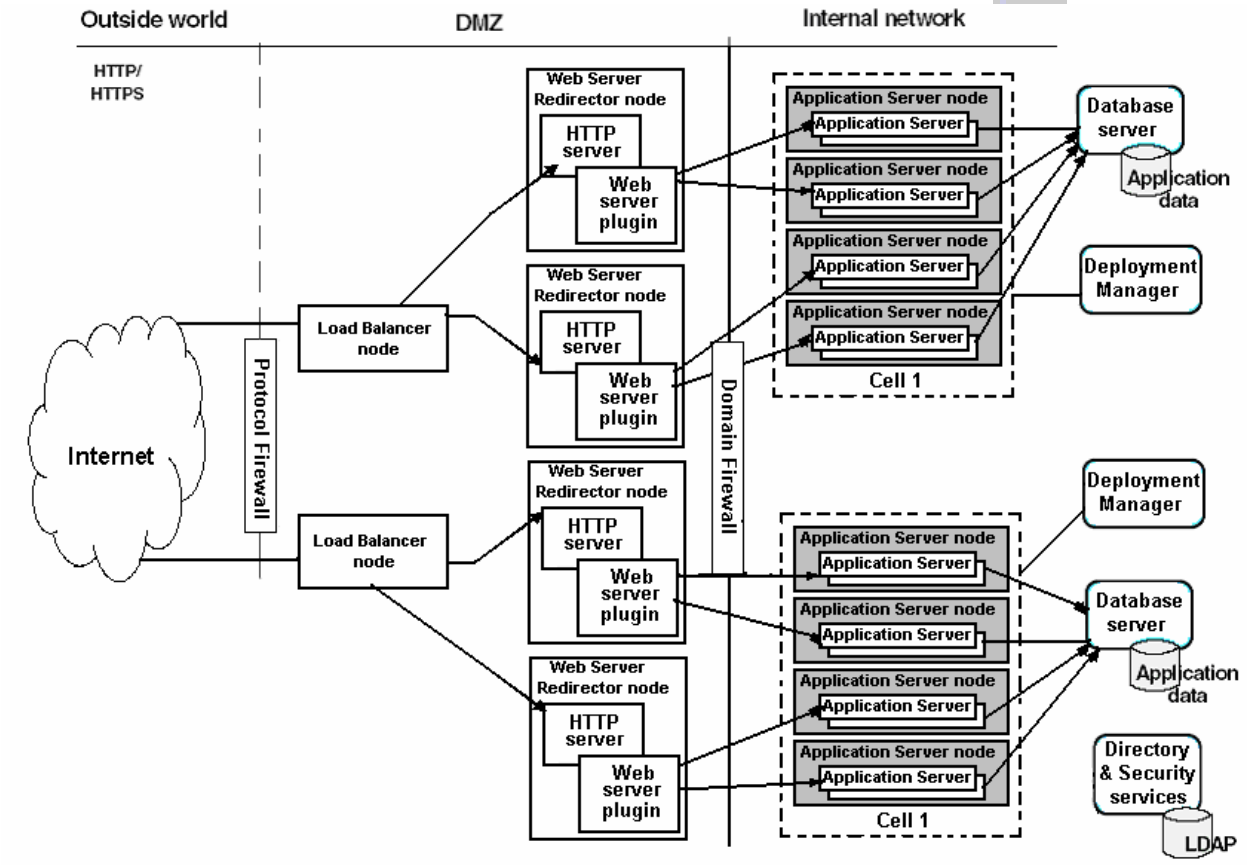
Hosting members of multiple application server clusters across one or more machines has the following disadvantage:

- More complex maintenance

  Application servers must be maintained on multiple machines.

12 Combined Topology

A combined topology utilizes the best elements of the other topologies.



This topology combines elements of several different basic topologies:

- Two WebSphere cells.
- Two load balancer nodes.
- Two HTTP servers for each cell with the Web server plug-in.
- Four application server machines for each cell.
- The use of application server clusters for both vertical and horizontal scaling.
- Each machine hosts two cluster members; in practice, the number of cluster members is limited by the computing resources of each node.
- Two database servers for each cell; these servers host mirrored copies of the application database.

12.1 Advantages

This topology is designed to maximize throughput, availability, and performance.

- Having more than one load balancer node, HTTP server, application server, and database server in each cell eliminates single points of failure.

- Multiple cells provide both hardware and software failure isolation, especially when upgrades of the application or the application server software are rolled out..

- Horizontal scaling is done by using both application server clusters and the IP sprayer to maximize availability and eliminate single points of process and hardware failure.

- Application performance is improved by using several techniques:
  - Hosting application servers on multiple physical machines to boost the available processing power.
  - Creating multiple smaller cells instead of a single large cell. There is less interprocess communication in a smaller cell, which allows more resources to be devoted to processing client requests.
  - The use of clusters to vertically scale application servers on each node, which makes more efficient use of the resources of each machine.

- Applications with this topology can make use of several workload management techniques.
  - The use of the Network Deployment workload management facility to distribute work among clustered application servers.
  - The use of the load balancer to distribute client HTTP requests to each HTTP server.

- Users will notice an interruption only when an entire cell is lost. If this occurs, the active HTTP sessions are lost for half of the clients. The system can still process HTTP requests, although its performance is degraded.

12.2 Disadvantages

The combined topology has the following disadvantage:

- Multiple cells require more administration effort, because each cell is administered independently.

  This problem can be reduced by using scripting to standardize and automate common administrative tasks.