Sun microsystems

SUN'S PATTERN-BASED DESIGN FRAMEWORK

THE SERVICE DELIVERY NETWORK: A CASE STUDY

Mikael Lofstrand, Sun Client Solutions
Jason Carolan, Sun Client Solutions

Sun BluePrints™ OnLine — April 2006

Part No 819-6319-10
Revision 1.0, 3/29/06
Edition: April 2006

Please
Recycle

TABLE OF CONTENTS

The Service Delivery Network: A Case Study

This Sun BluePrints™ Online article presents a case study that demonstrates using Sun's Service Delivery Network (SDN) to design the network architecture for an example secure e-mail application. SDN is Sun's approach to designing dynamic, service-optimized network architectures.

This article consists of the following sections:

For more information about SDN, including explanations of many of the concepts and terms mentioned in this article, see the Sun BluePrints article entitled *Sun's Pattern-based Design Framework: The Service Delivery Network*, by Mikael Lofstrand and Jason Carolan (Sun BluePrints Online—September, 2005), which is available at http://www.sun.com/blueprints/0905/819-4148.html.

About This Case Study

Secure messaging has emerged as a core IT service. Most organizations today rely upon e-mail as a mission-critical application that serves key business processes and transports proprietary and confidential business information among authorized users. The case study in this article shows how to use the SDN approach to guide the design of a secure, service-optimized network architecture for an example secure e-mail application. Secure e-mail was chosen for this case study because it is a familiar application that is relatively simple to describe and understand, allowing the reader to focus on the use of SDN rather than the details of an application. Note, however, that the SDN approach can be used to design network architectures that support almost any kind of application or service.

Note – Although this article describes a sample secure e-mail application, its purpose is to focus—not on the details of the application layer, but—on the process of designing a dynamic network infrastructure that will support the requirements of secure messaging services. The case study therefore represents an implementation *strategy* for delivering the messaging services. Application details—such as traffic load patterns, message volumes, or the average size of attachments—are outside the scope of this article.

The example secure e-mail application described in this article allows authorized end users to securely access e-mail messages—over the Internet or inside the corporate Intranet—using an e-mail client or a Web browser. From a security perspective, a traditional solution might consist of two separate

environments (Internet and Intranet) in a highly available deployment. In such a deployment, the message store servers would be shared to provide access to the same, centralized e-mail boxes, regardless of where and how clients would access them—via the Internet or Intranet. However, this approach becomes complicated due to the need to manage many individual components (such as service instances, networking devices, security devices, and so on). In addition, this approach might make tracking the costs of hardware, operations, and software more difficult.

Using the SDN approach, organizations can reduce costs—for example, by consolidating software instances and hardware, and sharing the infrastructure—while still providing a highly secure solution. Note that this particular implementation strategy is just one approach; your organization's specific needs might be better served by a different strategy.

The rest of this article walks through the process of designing the network architecture for the example secure e-mail application, describing the relevant activities for each phase of the process. For an introduction to these phases, refer to the *Sun's Pattern-based Design Framework: The Service Delivery Network* article at http://www.sun.com/blueprints/0905/819-4148.html.

Phase 1: Context

The Context phase involves the following activities:

- Define the Business Requirements
- Define the Functional Requirements
- Define the Service-Level Objectives

Define the Business Requirements

The SDN design process begins by defining the service to be delivered, the target consumers of the service, the business requirements and motivators for the service, and other high-level business issues. In the example secure e-mail application described in this article, the following business requirements were defined:

- The new messaging platform must be aligned with the organization's overall IT strategy and IT infrastructure.
- The e-mail service must be accessible to authorized users in a secure manner over both the Internet and the corporate Intranet.
- Users must be able to use an e-mail client or a Web browser to access their e-mail.
- The e-mail service must be highly available, able to integrate and interoperate with the existing IT environment, and able to be managed centrally and remotely.
- The e-mail service must use traditional open protocols, such as SMTP, IMAP, POP3, and HTTP(S).
- The organization's IT departments must be aligned with the business units to meet their needs.
- The IT services must allow for shared infrastructure and resource pools.
- The IT services must be reliable and secure business tools that increase users' ability to be more productive and efficient.

- The solution must increase automation to minimize operational complexity and lower risk as well as long term cost.

Define the Functional Requirements

The following functional requirements were defined for the new secure e-mail solution:

Component	Description
Messaging mail proxies	Handle inbound and outbound message transfers.
Messaging server	Server components to which users connect for access to their mailboxes.
Directory servers	Stores all directory information, such as the location of users' mailboxes. Also used for writing information into the directory.
Mailbox Store	Stores the contents of users' mailboxes.

Note that this article addresses only the core messaging components described in the above table.

Define the Service-Level Objectives

Service Level Objectives (SLOs) are the measurable factors that are used to determine the success or failure of efforts to meet Service Level Agreements (SLAs) for a given solution. These form an SLA. An SLA is measured on the higher level business service, but needs to be considered when designing the network platform. In general, applications and the other components of the infrastructure platform should also be robust and support the SLOs in a systemic fashion. For a detailed discussion of SLAs, see the Sun BluePrints OnLine article titled *Service Level Agreement in the Data Center* by Edward Wustenhoff (April, 2002) at: http://www.sun.com/blueprints/0402/sla.pdf

The following Service Level Objectives are fictional and shown as examples:

SLO	SLA	Network Metrics	Description
Availability	99.99% (measured at the system data interface, or SDI)	Network availability should be 99.99%	Unplanned downtime should not exceed 50 minutes a year of service due to network outages in the data center. Services should be scaled to support scalability metrics plus provide sufficient capacity to guard against SLA failure.

SLO	SLA	Network Metrics	Description
Scalability	1.5 million total subscribers	Must support a maximum of 60,000 active subscribers and 20,000 concurrent subscribers.	The network should support increases in the number of users without needing to be redesigned. The service should be scaled to the number of instances required to support the required load.
Latency	Users should experience no greater than five-second response time for e-mail access (via Internet)	Recommended maximum latency in the data center network should not exceed 300 milliseconds.	The network should use only wirespeed networking devices with sufficient capacity and full-duplex operations.
Performance	1,000 peak messages per second	The network should support Gigabit per second (Gbps) speeds between service domains and the service delivery interfaces. The estimated required network capacity for 1,000 peak messages per second is approximately 500 Mbits/sec.	Network devices should support Gigabit per second bandwidth on each port, and uplinks to the system data interface (SDI).

Phase 2: Forces

Forces define the relevant *goals* to achieve (such as high security and availability) in a service-optimized network design, as well as the pertinent *constraints* to work with (such as costs or standards compliance).

The Forces phase involves the following activities:

- Define Design Goals
- Define Design Constraints
- Assess Trade-offs

Define Design Goals

The design goals of the architecture must be defined so that they are inherent in the solution and support the SLOs and SLAs. While SLOs and SLAs define metrics for measurable values, design goals are defined to provide certain systemic qualities in the final solution. As an example, the design goals may differ from the SLOs, as some of the support for SLOs might be provided by product selections.

The following table describes the key design goals for the secure messaging solution.

Design Goal	Description
Availability	24x7 accessibility is required. Data partitioning and redundancy across multiple sites are required for providing disaster recovery in the event that a site failure occurs.
Security	Users must be authenticated and messages/attachments must be transmitted securely over the Internet. The solution must meet regulatory compliance requirements regarding e-mail communications. External networks must be considered, such as data transport between sites, routing constraints at the entry level of the data center, and so on.
Scalability	The solution architecture must be scalable to support an increasing number of e-mail users, as well as increasing volumes of message loads (number of messages plus the cumulative size of e-mail attachments). It might be possible to scale some requests horizontally by distributing the requests to multiple, similarly-configured systems. Some software components might scale up to a certain number of CPUs in a single system only (a good reason to use a horizontally scalable architecture).
Standards Support	ISPs offer a basic messaging solution for global usage with standard communication interfaces, such as SMTP, IMAP, and HTTP for millions of subscribers.
Manageability	Administrators must be able to easily and centrally manage all components of the solution.

Define Design Constraints

The key design constraints for the secure messaging solution include:

Design Goal	Description
Cost	Initial deployment and maintenance costs must fall within the budget allocated for this project.
Schedule	Initial deployment must be complete by the organization's deadline date.
Interoperability with Existing Environment	The solution must operate in, and integrate with, the existing IT infrastructure.
Sharable Components	The solution must be able to share infrastructure components in order to reduce costs as well as to provide a flexible, scalable solution that allows for changes "on the fly."

Assess Trade-offs

Trade-off decisions reconcile various forces that are in conflict. For example, achieving high availability via 100% redundancy for all solution components might cost more than the amount budgeted for the project. In this case, the solution architect might decide not to deploy certain components redundantly (to bring down costs) while still minimizing the risk of system downtime. Other trade-offs might include choosing a particular component despite a lack of certain features or functionality, or choosing between on-site, off-site, or remote support.

Trade-off decisions can be very complex. For example, suppose a decision involves choosing between two load balancing switches, one of which offers virtualization technology for routing, while another does not. In this case, virtualization technology means it is possible to virtually divide the load balancing switch into many logical load balancing switch partitions in the same physical unit. Each of the virtual switches has its own load balancing function, its own routing engine, and its own allocated memory as an example. The effect of virtual switching technology increases security and minimizes operational complexity by limiting—or, in some cases, even eliminating—the need for routing between the IP networks in the different

virtual switches and virtual routers. If there is no routing possible between networks, network filtering between those networks can be minimized or possibly eliminated.

If the application switch does not provide virtualized routing, it cannot pass traffic between security zones in a secure manner without network filtering. Adding network filtering to the design increases the solution complexity, because it requires additional configuration effort and precise tuning. On the other hand, an application switch with the virtualization technology allows traffic to be passed between the virtual routers using load balancing technology, which means that no routes are configured between the virtual routers, and therefore only the configured network traffic can pass.

Phase 3: Strategy

The Strategy phase involves the following activities:

- Select Products
- Select Building Blocks
- Select Patterns
- Design the Logical Architecture

Select Products

To ensure that service requirements are met, product evaluation and selection should bear in mind the context and forces that have been clarified in the previous two SDN phases. The software selection process is out of scope for this article, but as an example, this case study focuses on Sun Java Enterprise Systems software. In order to better understand the traffic flow in the design and the network services that need to be configured, this activity also defines the service relationships that are needed to build the solution.

The following table shows the mapping from the functional requirements (specified in “Define the Functional Requirements” on page 3) into component-based requirements based on functionality provided in the Sun Java Enterprise System products.

Service Component	Description
Message Transfer Agent (MTA) in	MTA inbound handles incoming messages. MTA outbound handles outgoing messages.
MTA out	
Messaging Multiplexor (MMP)	Component to which a user connects for IMAP or POP3 access to their mailbox.
Messenger Express Multiplexor (MEM)	Enables Web mail access for users.
LDAP Replicas	Store a copy of the master directory for applications to read information in the user directory.
LDAP Masters	Master directory that stores all directory information, such as where the users’ mailboxes exist. They are also used for writing information into the directory.
Message Store	Where users’ mailboxes are stored.

Once installed and functional, a service component becomes a service instance. The functional service component may be implemented as several service instances. Note that this example covers only the above e-mail application components.

Defining the relationships among the service components is a step that, on the surface, might seem to be excessive for conventional network design. In a traditional data center network, all data traffic is allowed on the network by default, which requires adding extra security components to the data center network, as security is a concern.

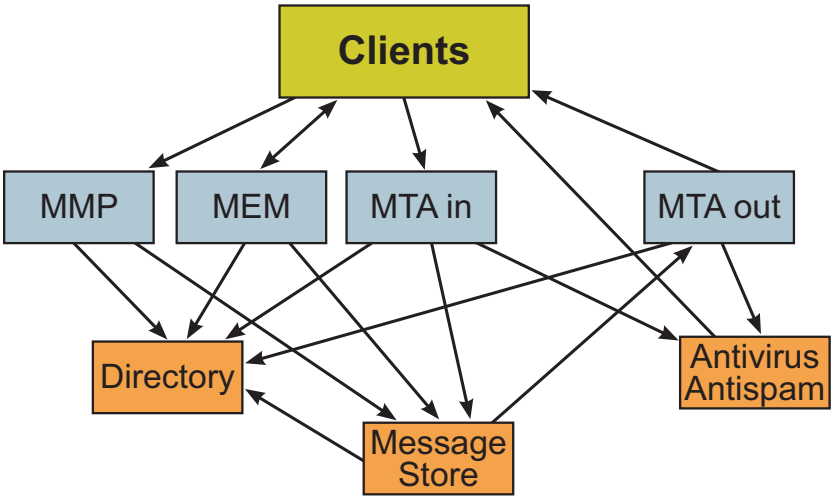

Figure 1. Service Relationships

The SDN approach, however, is to make the data center network secure by default and allow for service component communication as needed. This means that, for two different service components to be able to communicate with each other, it is necessary to configure the ISR (Intelligent Service Routing, described below) to distribute or route the traffic from the source to the destination. ISRs can support several modes of operation, such as routing, load balancing, redirection, virtualization, and so on. An ISR will therefore be able to ensure valid communication between service domains (see “Define the Service Domains” on page 9).

The following functionality was required in the application switch technology.

Feature	Description
Virtual Switching Technology	Provides additional security and compartmentalization per virtual switch.
TCP Termination	Provides deep packet inspection functionality.
SSL acceleration	Provides the ability to terminate SSL encryption in the application switch, as well as to re-encrypt sessions on the server side.
Load Balancing	Will distribute packets to the intended destination.
Link Aggregation	Will allow for grouping several physical links into one logical link.

The Sun Secure Application Switch, which provides all of these features, was chosen as a core component in the network architecture design.

The service components that are to communicate with each other will use a set of network protocols and TCP/UDP ports on the network. These network protocols and TCP/UDP ports must be identified, as they are defined in the load balancing switches when configuring the virtual services. This means, for instance, that when a client request comes into the SDNA via the service delivery interfaces and reaches a service access IP address, it is also matched with (as an example) a TCP port. This might be an IMAP request that is using TCP port 110 by default. If the IP address and TCP port do not match, then the request will be dropped and the session closed. The virtual services to be configured are the service instances that are deployed using the service components. All services that allow for virtualization will be virtualized and accessed within the service domain in which the client resides. The virtualization of the services is handled by the load balancing switches, which together provide the functionality called Intelligent Service Routing (ISR). The ISR provides the communication between the various service domains.

Note – It is important to revisit the defined software components after each of the following steps to make sure that conflicts with the business or operational requirements have not been introduced into the design.

Select Building Blocks

Building blocks represent the smallest, irreducible component within the pattern or architecture. In the SDN approach, building blocks include the modules, service instances, and logical execution containers. Once the forces have been clarified and trade-off decisions made (see “Phase 2: Forces” on page 4), you can begin to design the layout of the logical modules that will subsequently be used to help define the initial high-level design of this messaging architecture.

The following building blocks need to be defined:

- Service Domains
- Service Modules
- Service Delivery interface
- Distribution Modules
- Integration Security Module
- Service Security Modules
- Domain Security Modules

For more information about these building blocks, refer to the *Sun's Pattern-based Design Framework: The Service Delivery Network* article at http://www.sun.com/blueprints/0905/819-4148.html.

Define the Service Domains

Service domains provide host connection networks for the service instances. The segregation of the service components into service domains provides controlled network security, as well as flexible control over the data flow. In addition, the service segregation allows the possibility for load balancing, virtualization, and automation. Service domains are groups of similar services, and will include one or more service instances.

Service domains can be categorized by the following component types:

Component Type	Description
End-User Service Component	Services provided directly to an end user. Examples include a Web site, the ability to send e-mail, and Usenet news.
Supporting Service Component	Services that directly support end-user services. One example is an application server that provides dynamic content for a Web site or an e-commerce application. However, the end-user service is the service access point—users do not directly initiate connections into a supporting service.
Infrastructure Service Component	Services that ease internal operation and support, such as system and network management services. One example is an internal DNS service. Infrastructure services should not directly interact with end-user services.

Define the Service Component Types

The following table summarizes the services and the service component types in the secure messaging application.

End-User Service Name	Service Scope	Network Service Component	Service Component Type
E-Mail IMAP	External	Message Relay (Ext)	End-User Service
E-Mail / HTTP	External	Message Relay (Ext)	End-User Service
E-Mail IMAP	External	Message Relay (Int)	End-User Service
E-Mail / HTTP	External	Message Relay (Int)	End-User Service
	Internal	Directory	Supporting Service
	Internal	Message Store	Supporting Service

It is also important to define security zones and management domains in the secure messaging application. A detailed discussion of these topics is outside the scope of this article. Briefly stated, a security zone can be described as a grouping of service domains with similar security requirements, which makes it easier to determine the grouping of management domains or the need for additional security modules.

Mapping Service Components to Service Domains

The following figure shows an example of the logical mapping of service components to service domains.

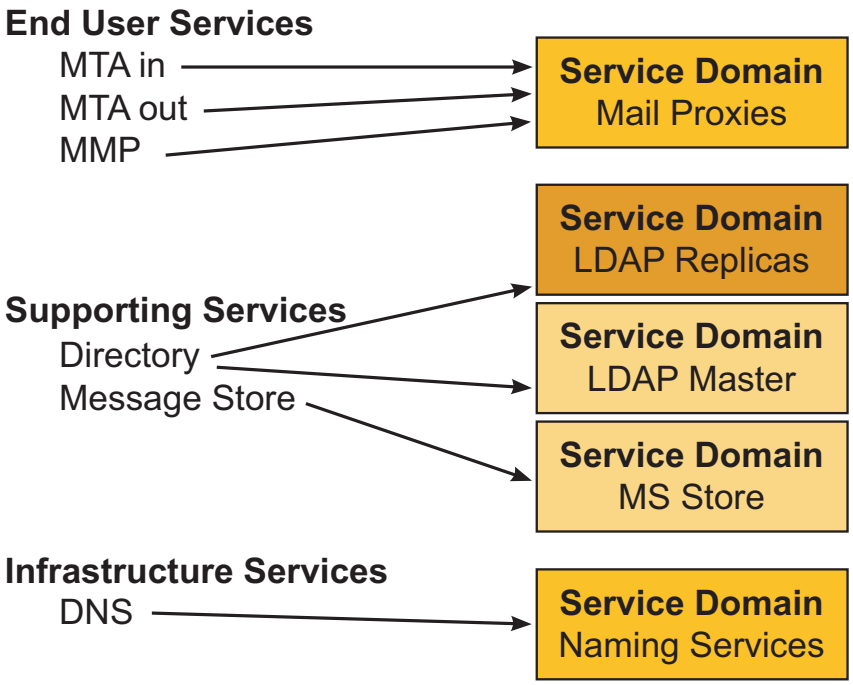

Figure 2. Mapping Service Components to Service Domains

These service domains are compartments of broadcast domains that should be secure by default. This implies that no network communication is possible into—or out of—the service domain at initial setup.

The domain security module could consist of a transparent (bridging) firewall for the MS Store and the LDAP Master. However, this example assumes that virtualization technology (instead of a firewall) will provide the necessary network security barrier between the service domains. Different security zones will be in separate virtual switches.

The service domains are combined to build the service module for Internet messaging. In addition, there is a separate service module to handle Intranet messaging. The virtualization technology allows for a logical separation of the two messaging solutions. Note that, even though two logical service modules are configured, these two service modules still share the same physical components (see Figure 3).
The service modules have different paths and different functional layers, even though they share physical components and service instances. The service domains therefore span the two service modules. Each service module has its own service access point, which is represented by an IP address on a specific IP network range per service delivery interface.

Select Patterns

A *pattern* is a generalized, reusable design solution to a recurrent network architecture design problem. This section describes the patterns used to construct the network architecture design for the secure messaging application. For a detailed description of SDN design patterns, refer to the *Sun's Pattern-based Design Framework: The Service Delivery Network* article at http://www.sun.com/blueprints/0905/819-4148.html.

Service Module Design Pattern

The service module design pattern provides the core hardware, software, and configuration components, consisting of network devices (such as load balancing switches and host connection switches), that are configured in a specific way to support service instances configured within service domains. The following figure shows side-by-side service modules, one delivering Internet services and the other delivering Intranet services, and both sharing the same physical components.

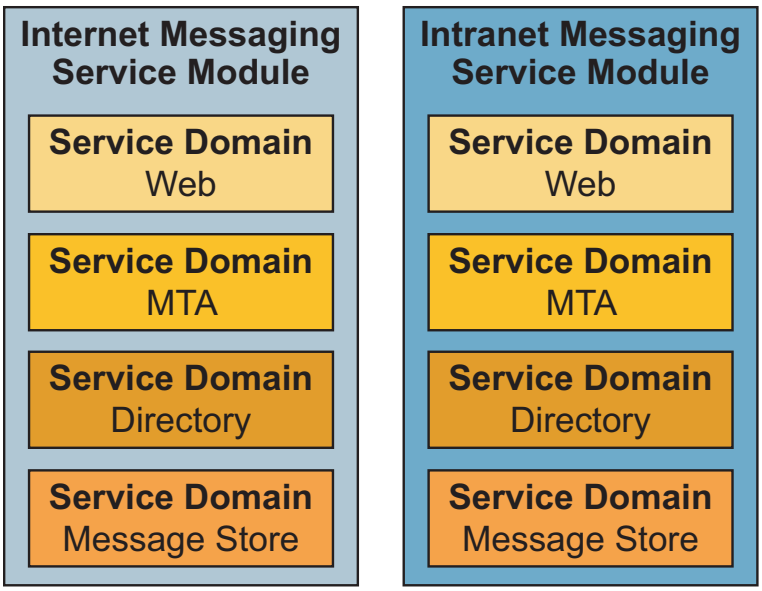

Figure 3. Logical Service Domains

Security is enhanced by utilizing the virtualization technology for network communication between the service domains. If another service module were to be added for a different kind of service (such as a financial service), there might be a need for an additional service security module for that particular service module. Such a service security module might include a high performance firewall appliance.

Multi-Service Module Design Pattern

Because this design has multiple service modules, service routing to each of the service modules is required. A distribution module can handle service routing, which provides a path from the service delivery interface to the service access point for each of the service modules. The multi-service module design pattern includes two or more service modules and a distribution module.

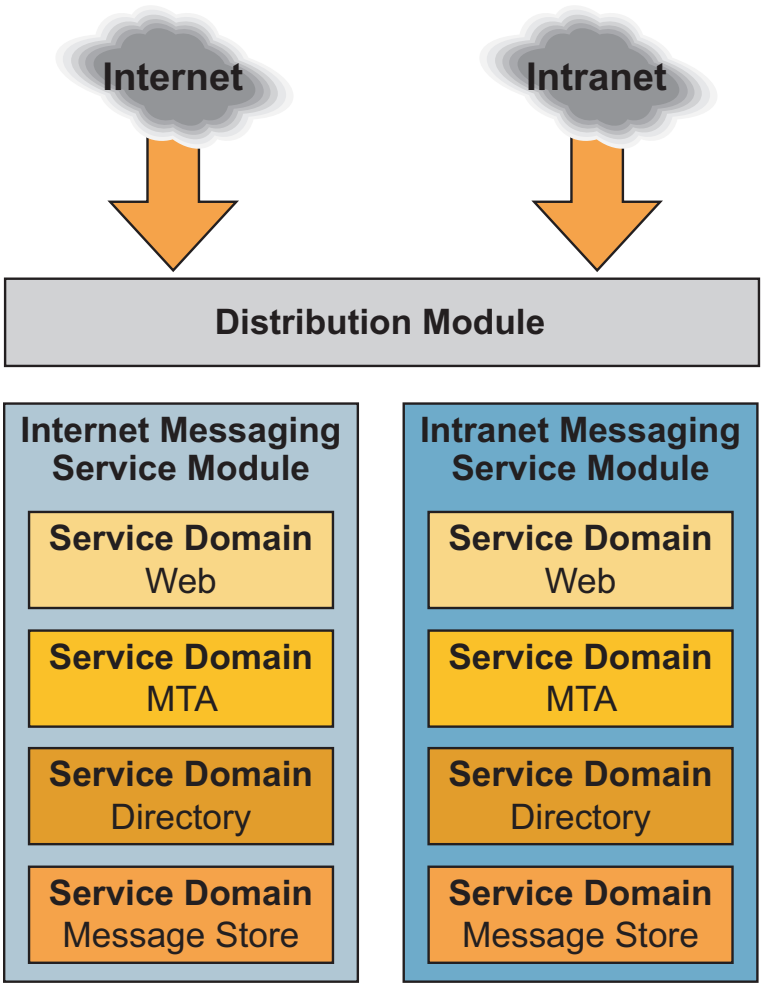

Figure 4. Multi-Service Module Design Pattern

Multi-Service Module with Integration Security Module

The following pattern extends the multi-service module pattern by adding an integration security module, which provides additional network-layer inspection before traffic enters the domain module.

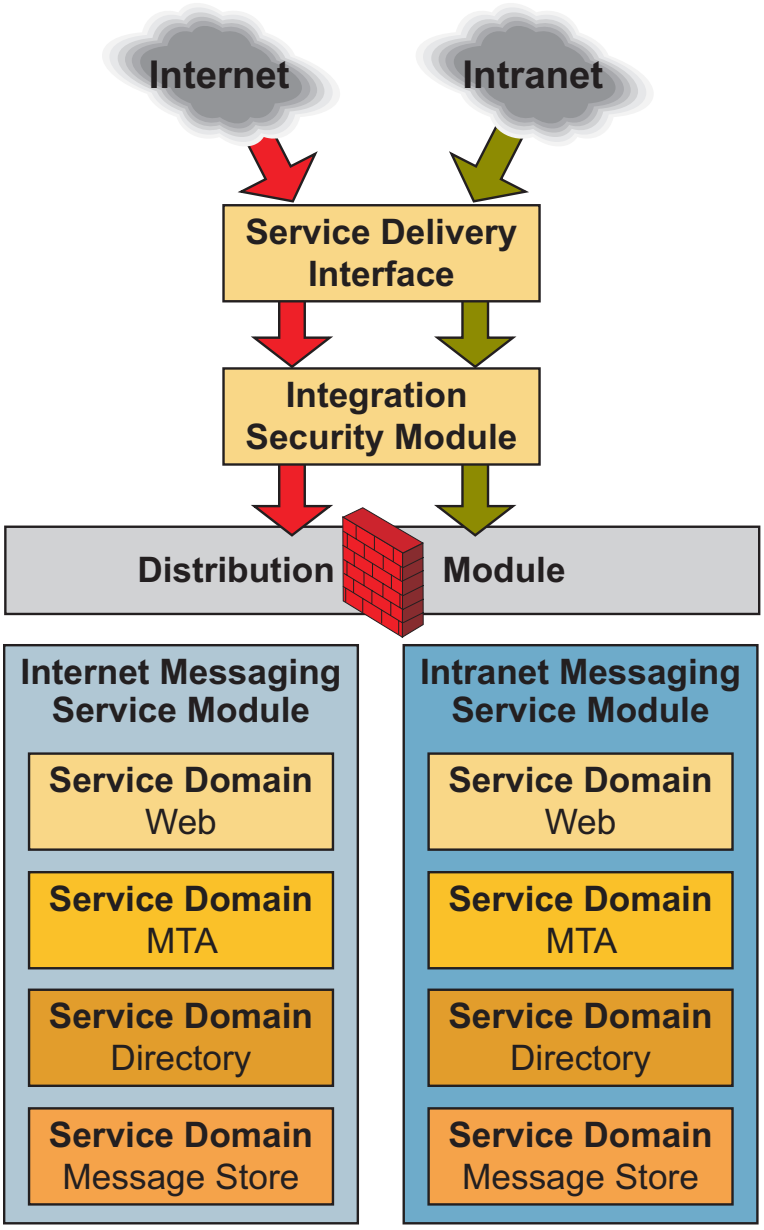

Figure 5. Multi-Service Module Design with Integration Security Module

In this example, the integration security module sits *below* the service delivery interface and provides network access security and logging, with secure network paths through the Distribution Module.

Design the Logical Architecture

Once the logical layout of the architecture has been defined, the foundation for the next step has been laid: designing the network for the business service.

The following block diagram shows the relationships among the service domains, Integrated Service Routing (which provides the network path between the service domains), service delivery interface, and secondary access networks (used for integration to legacy systems, if applicable).

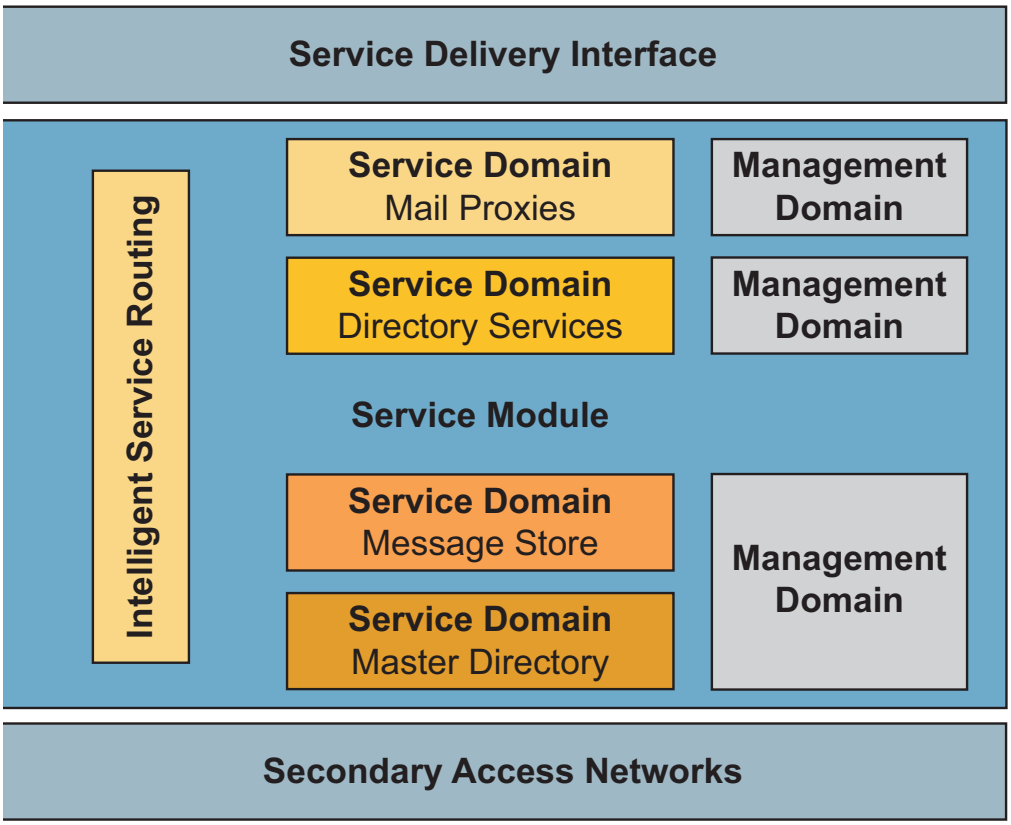

Figure 6. Logical Layout

In addition, this figure shows that the management domains are grouped for the service domains based on security requirements. If, for example, the message store and the master directory had the same security requirements, it would be possible to let those service domains share the same management domains.

The following figure looks like a more traditional logical diagram.

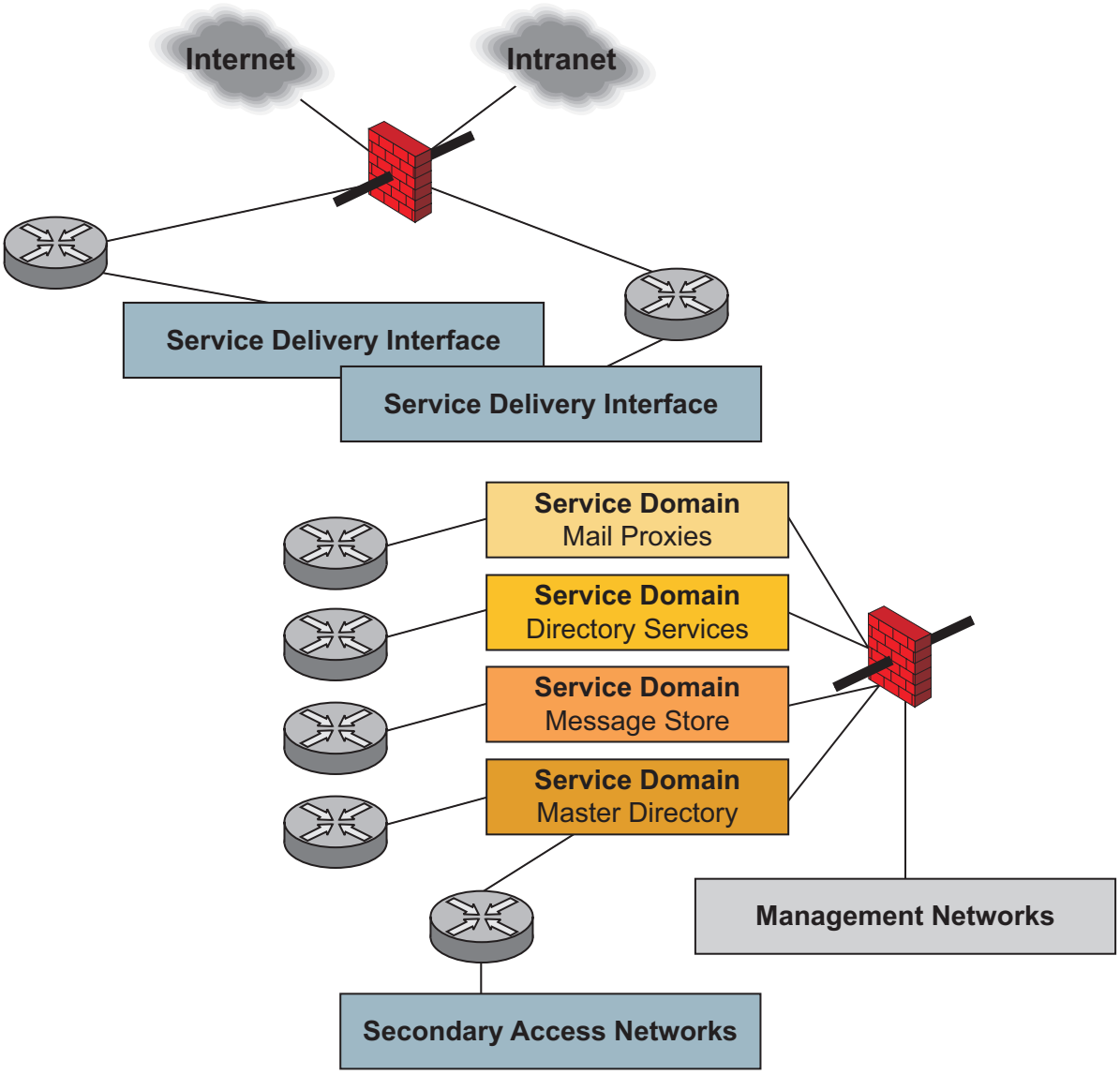

Figure 7. Traditional Logical Diagram

Phase 4: Solution

The Solution phase involves designing the physical and implementation architecture for the solution.

Gathering Additional Information

Before designing the network solution architecture, some additional information is required—the IP address space for each of the service domains, the service delivery interface, and the service access points. In addition, networking products that can match the requirements from a service perspective need to be identified.

To match all components with requirements in a composed solution, it is necessary to revisit the requirements definitions (described in “Define the Business Requirements” on page 2 and “Define the Functional Requirements” on page 3). For example, suppose a network design uses the Spanning Tree protocol to provide network Layer 2 redundancy. If a switch failure occurred, then it would have an impact on service availability for all services with a traffic flow through the impacted switches. This service might be unreachable for only three seconds, but—depending on the application—it might also break user sessions. Disrupted user sessions might have a larger impact on the business.

Assumptions

This phase makes the following assumptions:

- There are two sites that are to be used for redundancy in support of the high availability requirement for the secure messaging solution. These sites are within the range of fiber distance, which can extend a layer 2 network between the sites.
- All connections between network components are Gigabit Ethernet, except for the connection between the separate sites (which will be connected via DWDM).
- Just as for any deployments of a business service, there needs to be a separate test and development environment that replicates the production environment as closely as possible. Note that details of the test and development environment are not described in this article.
- The development compute resources must be able to be reallocated into the production environment upon demand without recabling.
- The development environment will not interfere with the production environment.
- Seamless failover should occur if a layer 2 switch fails.
- There should be minimal impact on service availability between client and service, including availability between end-user services and supporting services.

Solution Architecture

This section describes the highlights in the resultant design, based on the SDN approach, for the network solution architecture for the secure messaging application.

Layer 4-7 Configuration

The following figure shows an overview of the automated virtual platform.

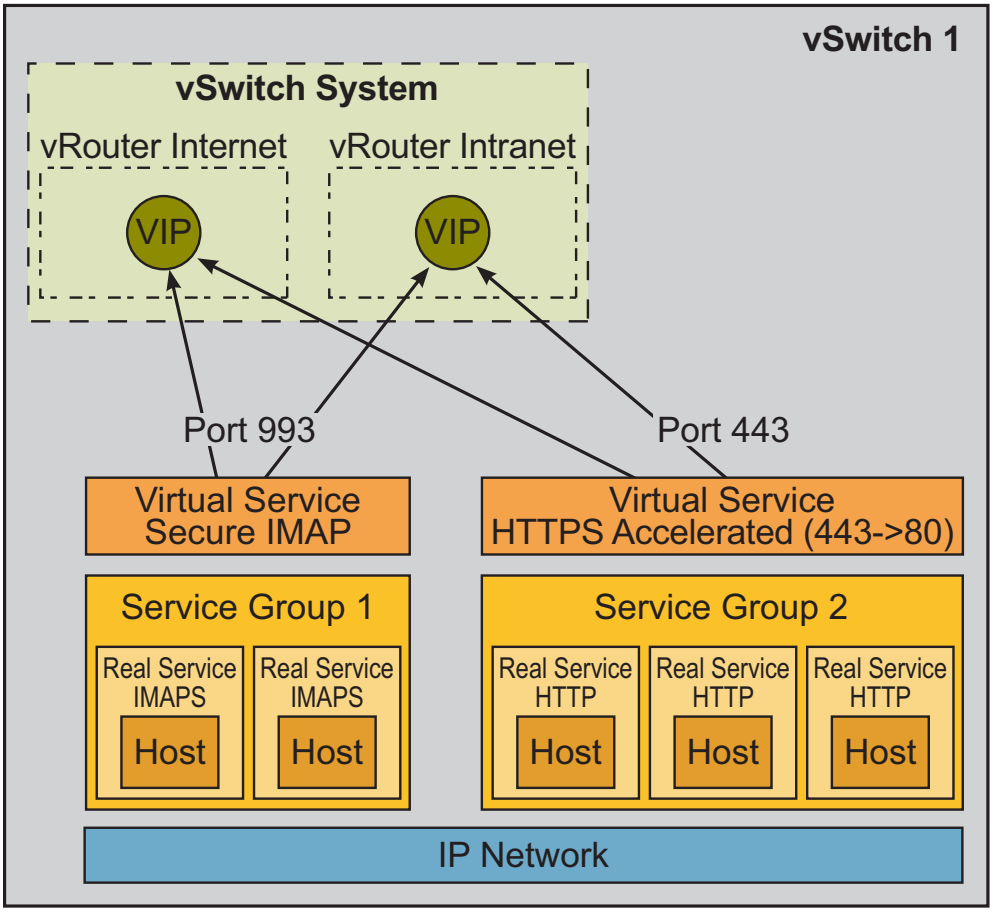

Figure 8. Layer 4-7 Configuration

This figure shows the layer 4-7 information on how to configure the Sun Secure Application switch (see http://www.sun.com/products/networking/switches/) for the end-user services for e-mail access via Secure Imap and HTTPs. In this example, all the hosts are on the same IP network. The live services, as shown in Figure 8, are the necessary network services (such as IMAP or HTTP) that are configured for each host. Note that multiple real services can be configured per host.

Health Checking

The real service command, used specifically for the Sun Secure Application Switch, binds a named host and port to a named service. A real service is put into a service group to be part of forming a virtual service. The service group is load balanced, which distributes the request load and provides a failover mechanism via health checking of the real service.

Health checking is based on inline traffic, with the ability to fall back to out-of-band health checking if the inline health checks fail. This approach means that the inline health checks make use of the live application traffic between a source (client) and a destination (server/service instance) to verify that

communication occurs between the client and the service. If this communication fails, the Sun Secure Application Switch disables the real service and does not re-direct traffic to it. In parallel, the Sun Secure Application Switch sends out-of-band, server health checks to the real service to determine when the service becomes available again, at which time it will bring the real service back into the group as active. In this case, a logging entry can be sent to a loghost for notification to the network operations team that this service has failed. The notification can also be achieved by using network management tools such as, for example, one based on SNMP.

Layer 3 Configuration

Virtual services are also configured for the service groups to provide virtualization of the services. The virtual IP addresses are the actual service access points. They are configured in the same virtual switch as the virtual services. However, the virtual IP addresses can be placed in any other virtual switch and virtual router within the same Sun Secure Application Switch.

The following figure of the layer 3 configuration shows how the networks are segregated.

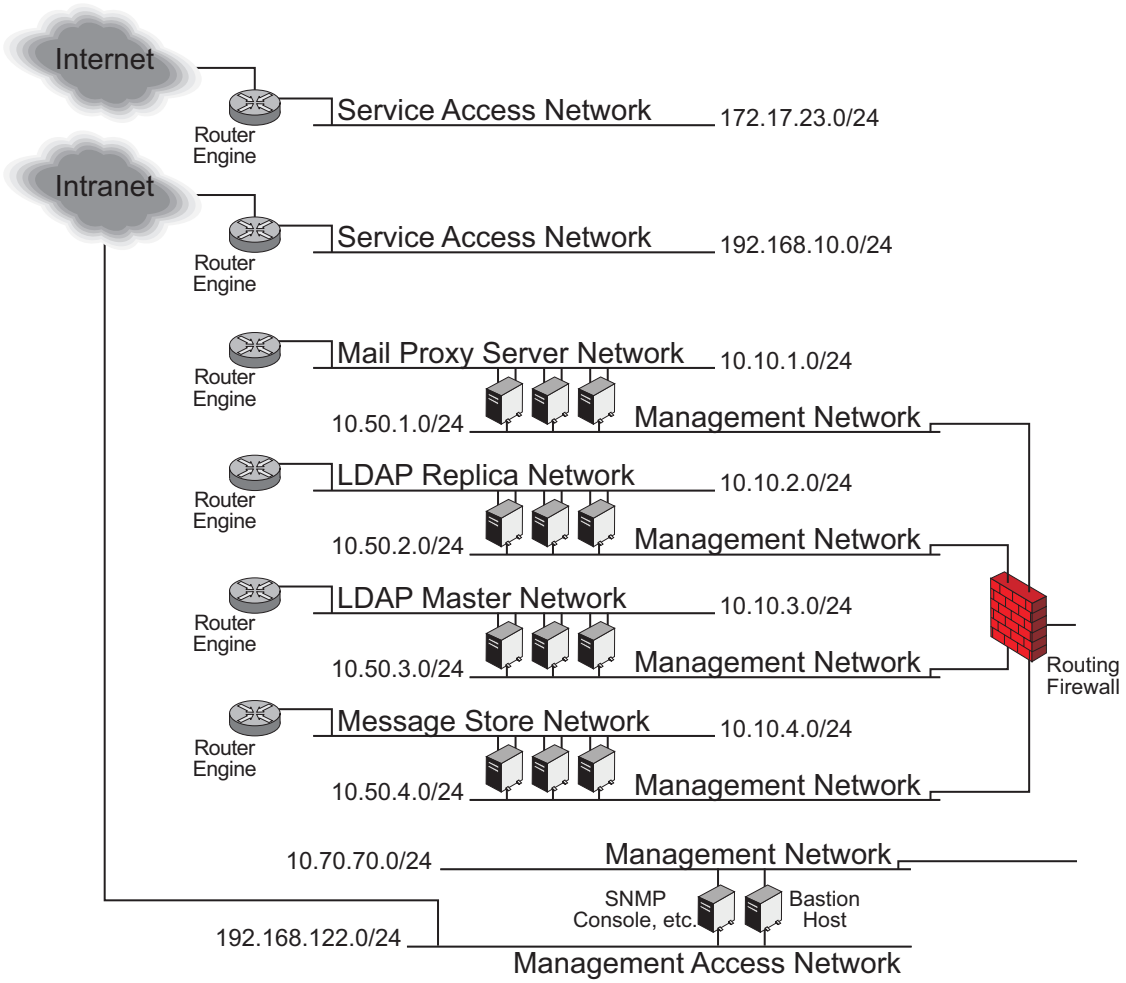

Figure 9. Layer 3 Configuration

Figure 9 shows how there are specific networks for network communication, which has a direct dependency on the delivery of the network service that serves the business application. In this figure, these networks are the IP ranges that have 10.10.X.0/24 networks. In this example, the servers reside on these networks and represent a service domain per network.

The two service access networks in Figure 9 have IP nodes that are the service access points. These can also be described as virtual IP addresses. They represent the only access point to the business application. There are no physical servers residing on these networks. These networks exist in the service delivery interfaces and provide the path into the service delivery network.

In Figure 9, the router engines do not represent connections between the routing engines, simply because there is no route between the routing engines. One might reasonably ask how a business application, which consists of many software components providing different services, can operate if no route exists between the networks in which the software components reside? The answer is that all of the routing engines are actually virtual routers in different virtual switches provided within the Sun Secure Application Switch. Each virtual router and virtual switch has its own tables and does not share this information with other virtual switches or virtual routers. The virtual routers can actually be configured to work together using static routes or routing protocols. The traffic will be forwarded internally in the Sun Secure Application Switch from the virtual IP address configured in the virtual service. The virtual service is configured in the virtual switch, where the networks for the physical servers resides, as shown in the layer 2 diagram (see Figure 11) later in this article.

What are the consequences of this design? Because there is no route between the different networks, no undesirable traffic can pass from one security zone to another. The only communication that will pass from one service domain to another service domain (configured in different virtual routers) is the communication, via the virtual IP address, that meets the criteria configured for that particular service. This means that Intelligent Service Routing must be configured.

Management Networks

Management networks exist in the 10.50.x.0/24 IP address network range and have separate interfaces in the servers, as shown in the following figure.

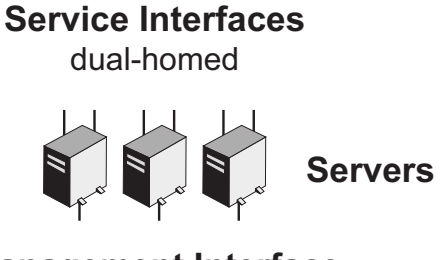

Figure 10. Separate Interfaces in Servers

These networks represent the management domains. They are connected to a firewall, which keeps them separated from each other to prevent unauthorized traffic between the management domains. The firewall

is also connected to another management network via the 10.70.70.0/24 IP network (see Figure 9). Two IP nodes exist between the 10.70.70.0/24 network and the 192.168.122.0/24 network: one is an SNMP console with two interfaces, and the other is a bastion host, which, in this scenario, is based on a role-based remote access server (such as the Appgate server from Appgate network security, described in the “Appendix: Management Domain Access with AppGate Network Security” on page 25).

The reason for having two interfaces on these IP nodes is to prevent *asymmetric routing*, which occurs when, for example, a server has a default route configured and there are two paths from a client to the server. Traffic might enter the server via one interface and leave the server via another interface, using a different route through the network. This can result in broken sessions if session state needs to be kept.

There will also be less control of the data flow if asymmetric routing occurs when there are multiple distribution layers that handle the intelligent service routing. This can occur, for example, if the client has routes into the server via multiple network paths. A route through the Service Delivery Interface, as well as via the management network and the source address of the client, is preserved in both paths all the way down to the server.

Virtual Routers

The following figure shows the various virtual routers that are configured within the Sun Secure Application Switch.

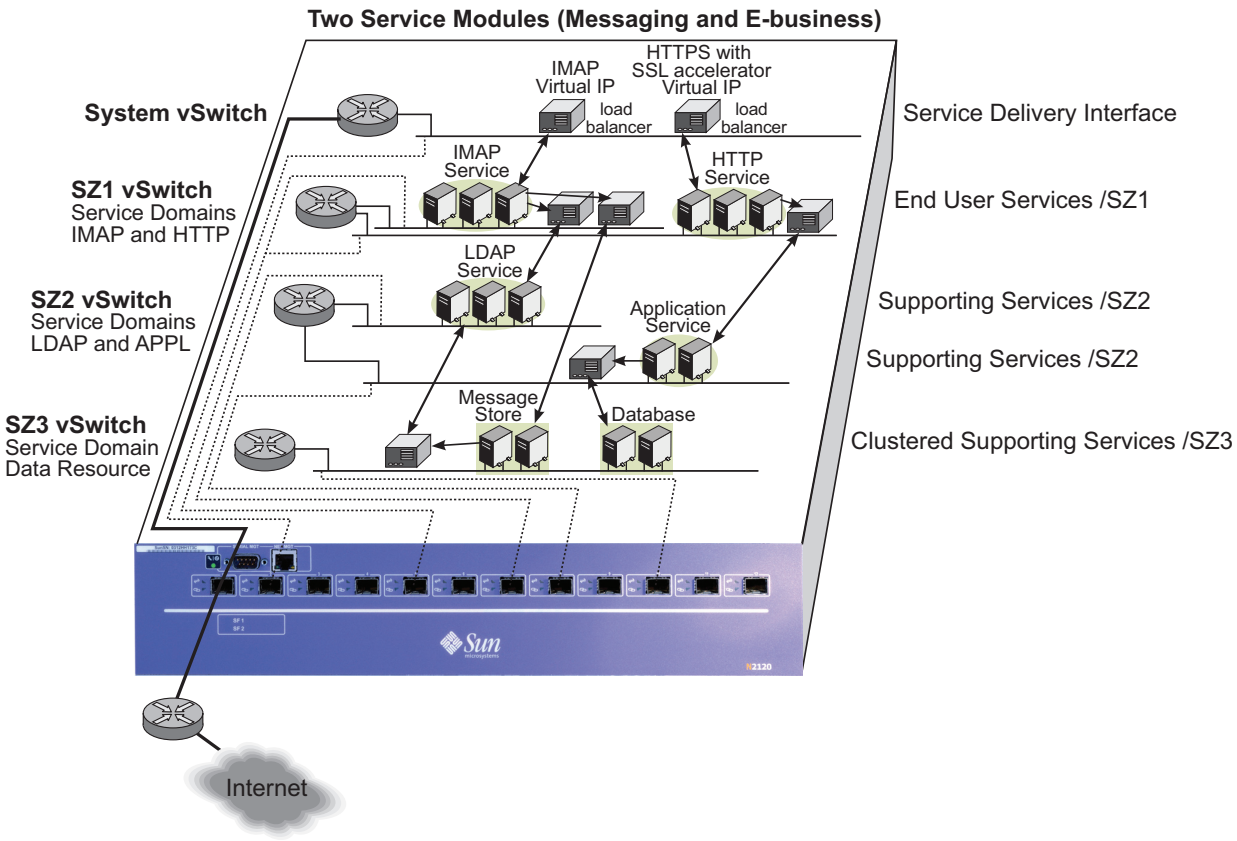

Figure 11. Layer 1-2 Configuration

Used in conjunction with Figure 9, this figure provides a better understanding of where the IP networks are actually configured, and how it is possible to implement the separation of Internet and Intranet messaging services explained earlier in this article.

Physical Network Component Design

The following figure shows the physical network component design layout for the production environment.

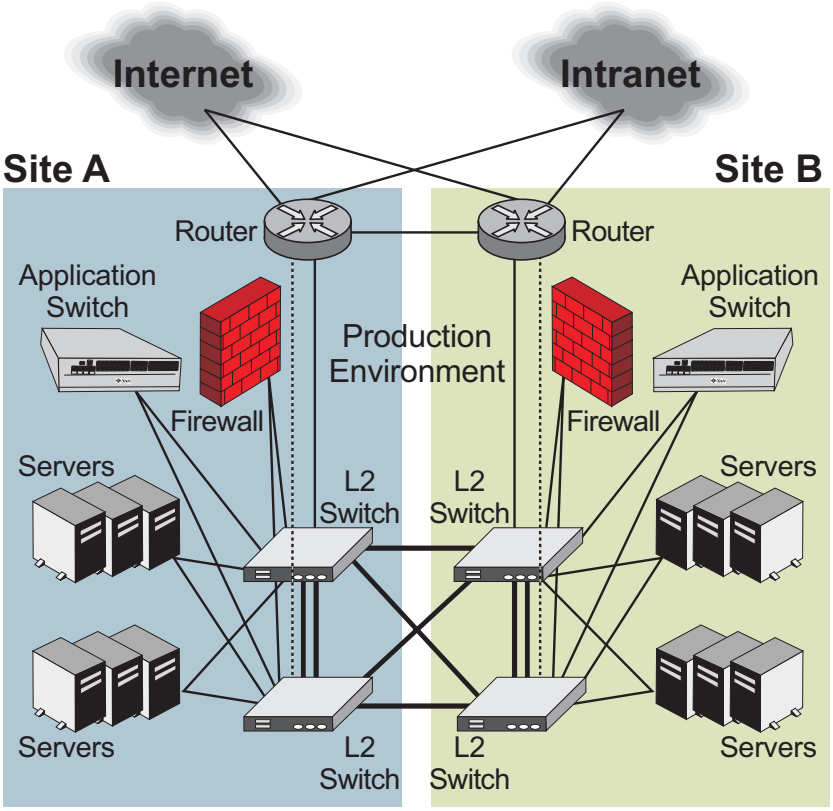

Figure 12. Physical Network Component Design for the Production Environment

This figure shows four chassis-based layer 2 switches in a fully meshed design in which all VLANs are present in all four switches. This design provides for soft cabling capabilities that allow cabling up a server once, without the need for subsequent recabling—even if this server is reprovisioned. The value in this design approach is that, even if the data center grows to exceed the capacity of these switches, it is still possible to add switches into the mesh and expand the existing VLANs out to the newly attached switches as well.

Note – The fully meshed layer 2 network can be implemented in several ways and should be carefully considered, using the desired qualities of service availability, vendor relationships, costs, and other relevant factors.

There are several ways to implement the fully meshed layer 2 network, each of which yields a different quality of service for availability.

- For example, if a standard such as the Spanning Tree protocol is used, then there will be a block of service availability during the reconvergence time if any switch in the mesh fails or is powered down or up. If VLAN tagging is used on the interconnects between the switches, and if a cable fails or is plugged

in or unplugged, then the same reconverge will occur, and all VLANs on that interconnect will be affected. This is the standard behavior of the Spanning Tree protocol.

- An alternative to Spanning Tree would be to use, for example, Split Multi-Link Trunking, which is a proprietary solution from Nortel Networks to solve the same issues. What is different is that the layer 2 network failover occurs in a sub-second, and the end user will most likely not be affected. Another advantage of this proprietary solution might be that the risk of broken sessions is lowered, depending on how the peripheral components are attached and how can they handle the failover.

The meshed Layer 2 network, in combination with the virtualization technology in the Sun Secure Application Switch, allows servers to be moved to a different service domain by a reconfiguration of the servers' representation in the network configuration. Servers will thereby inherit the attributes for the new role as specified for the members of the "new" service domain.

To be able to quickly adapt to changes in the infrastructure, network provisioning delivers the tool for automated deployment and modifications. Network provisioning provides scalable and consistent device configuration tasks that minimize the risk for errors and enable improved time to market.

By minimizing the manual operational tasks, it is possible to provide additional security by ensuring conformance with standard processes. Automated network provisioning allows servers to assume a new role for a different service in a different service domain or service module.

As an additional strategy, the following figure shows the physical network component design layout for the production environment, with the addition of a development and testing environment.

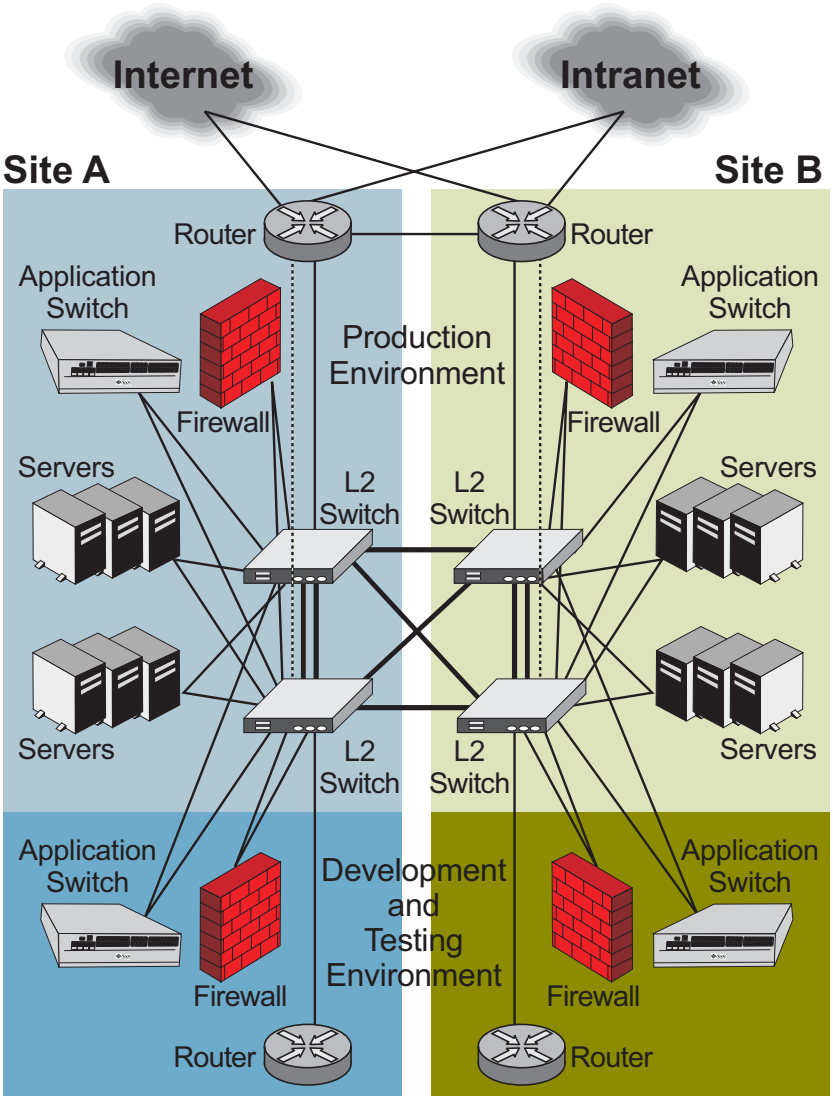

Figure 13. Physical Network Component Design Layout—Production and Development & Test Environments

In this case, the fully meshed layer 2 network is shared and can move servers between the various environments as needed. This approach can save the cost of expensive network devices as well as computing nodes. In addition, the layer 2 mesh—in combination with network provisioning—enables a smooth and controlled transition for a service to move from development to test to pre-production and, finally, into the production environment.

Appendix: Management Domain Access with AppGate Network Security

Some kind of bastion host should be implemented to minimize the possibilities of routing loss in the network, which can occur when there are several paths into a server. For example, an administrator can be on a LAN environment in the corporate Intranet, and he can access the servers via the administration network as well as using the messaging service, just like any other user. This might cause problems when the user accesses the server from different interfaces, and the server does not know exactly which path it should use to send the data packets back to the client. The problem occurs when and if the server sends the return packets via an interface other than the one from which it received the packets. This may break the session. The management access should thereby be distinguished in some manner, and the source IP address should be local or have a unique path.

Secure remote administration of a data center system can be a crucial issue. A security server can be installed that offers secure, role-based administration. Using the built-in rights management system, it is possible to define exactly which systems and administrative tools each administrator is allowed to access in the data center. All communication is encrypted from users' workstations and tunneled within a single TCP connection to the security server to facilitate firewall traversal.

The data center administrator can define exactly what authentication method should be required for each local administrator (for example, based on who the administrator is and from where they connect) to demand the use of certificates or smart cards from all external networks. It is also possible to make sure that only certain systems be allowed for remote administration, and/or that a personal firewall running on the client system must enforce a specific rule set, such as disabling all Internet traffic. All these efforts are geared toward guaranteeing secure remote administration of the servers.

The security server can easily give administrators access to administration tools running on individual application servers. In many cases, secure single sign-on can be enforced—the user's identity is passed on from the security server to the local server. The security server can provide access to different parts of a web server (URL filtering) to give local administrators access to different parts of the server.

The security server can consist of redundant systems in which each server has multiple network interfaces to offer network redundancy. The client software automatically connects to an available server in the cluster, completely transparently to its users. The security server also keeps detailed logs over all accesses, and can be configured to send alarms (for example, using Syslog or SNMP) to external systems.

Security server functionality includes:

- Role-based administration. Administrators can reach and administer different parts of the application server system.
- Support for most authentication methods on the market. The method can be selected based on security needs and the network infrastructure.
- User activity logging and traceability.
- The system can have multiple network interfaces for incoming traffic and can be clustered for high availability.

- Support for arbitrary management protocols to application servers.
- Automatically updated Java-based client software. Runs on virtually all types of systems.
- All traffic securely tunneled through one TCP connection, which makes it easy to traverse firewalls and proxies.

References and Related Sources

Sun's Dynamic Infrastructure for Web Services

The SDN is part of Sun's more comprehensive Dynamic Infrastructure for Web Services, which emphasizes dynamic data center service architectures, with increasing capabilities in virtualization and automation. The Dynamic Infrastructure employs optimization design patterns across a broad range of IT infrastructure elements, including servers and storage, operating systems, grids, middleware, and shared services such as directory and the J2EE™ software, as well as application execution and messaging. The Dynamic Infrastructure is based on core Sun technologies such as the Solaris™ 10 Operating System, Sun Java™ Enterprise System (JES), N1™ Grid Technology, and the Sun™ Secure Application Switch hardware. For more information about the Dynamic Infrastructure, see:

http://www.sun.com/servers/dynamicweb/

Sun's Service-Oriented Architecture

Sun's Service-Oriented Architecture (SOA), an integrated software infrastructure and design approach based on industry best practices, complements SDN and the Dynamic Infrastructure for Web Services. SOA is an architectural style for building and composing software applications that use services available in a network. The SDN helps provide the communication fabric upon which the SOA is realized. It provides a foundation to build cost-effective, easy to manage services. For more information about SOA, see:

http://www.sun.com/products/soa/

Publications

- *Sun's Pattern-based Design Framework: The Service Delivery Network*, by Mikael Lofstrand and Jason Carolan (Sun BluePrints Online—September, 2005)

  http://www.sun.com/blueprints/0905/819-4148.html.

- Alexander, Christopher. *The Timeless Way of Building*. (Oxford University Press, 1979).
- Dyson, Paul and Longshaw, Andy. *Architecting Enterprise Solutions: Patterns for High-Capability Internet-Based Systems* (John Wiley & Sons, Ltd., 2004)
- *Toward Systemically Secure IT Architectures*, by Glenn M. Brunette (Sun BluePrints Online—February, 2006)

  http://www.sun.com/blueprints/0206/819-5605.pdf

- Carolan, Jason; Radeztsky, Scott; Strong, Paul; and Turner, Ed. *Building N1 Grid Solutions: Preparing, Architecting, and Implementing Service-Centric Data Centers*, which is available at the following URL:

  http://www.sun.com/blueprints/pubs.html

Web Sites

- Service Delivery Network Architecture Roadmap Services
  http://www.sun.com/service/sunps/architect/delivery/roadmap.html
- Architecture Services for Service Delivery Networks
  http://www.sun.com/service/sunps/architect/delivery/
- Service Delivery Network Architecture Implementation Service
  http://www.sun.com/service/sunps/architect/delivery/implementation.html
- Sun Dynamic Infrastructure for Web Services
  http://www.sun.com/servers/dynamicweb/
- Sun Secure Application Switch
  http://www.sun.com/n2000
- Sun Security
  http://www.sun.com/security

About the Authors

Mikael Lofstrand

Mikael Lofstrand is a Principal Engineer at Sun Microsystems. Mikael has spent most of his time at Sun developing solutions for various customers improving systemic qualities through virtualization technologies. Mikael has recently spent two years in Sun's Strategy and Architecture team working on improving efficiency for globalized data centers using various networking technologies. He is also one of the original authors of the Service Delivery Network Architecture, Sun's modular, data network architecture standard. Mikael often speaks at conferences world wide about the importance of network designs for a service-driven architecture.

Jason Carolan

Jason Carolan is a Technical Director and Principal Engineer at Sun Microsystems. He has spent over six years in Sun Professional Services and Solutions organizations, focused on developing solutions for Sun's customers, systems architectures, and improving architectural quality through patterns. Jason contributed to many of the early internal and external documents of Sun's N1 software. He has also been responsible for the design of the Service Delivery Network Architecture, CSO’s data center standard—a key example of modular architecture and reuse. He also speaks regularly at conferences throughout the world about network design, security, and the N1 Grid architecture. Jason lives in Tucson, Arizona, where he enjoys the amazing sun.

Acknowledgments

The authors would like to thank Glenn Brunette, Roy Pillay, and Tomas Olovsson for their contributions to this article.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject at `http://docs.sun.com/`.

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at:
`http://www.sun.com/blueprints/online.html`