

Web-Infrastrukturen mit dem SAP NetWeaver® Application Server

Oliver Nocon, Volker Zirkel



Inhalt

1	Einführung	5
2	Web-Infrastrukturen	9
2.1	Technische Anforderungen von Webapplikationen	9
	Transaktionalität	9
	Skalierbarkeit	9
	Performance	10
	Sicherheit	10
	Stabilität	11
2.2	Web-Infrastruktur-Szenarien	13
	Typische Webzugriffsszenarien	13
	Klassifizierung von Webzugriffsszenarien	14
	Ist der Zugriff aus dem Internet auf ein internes System sicher?	15
	Lässt sich dieselbe Infrastruktur für internen und externen Zugriff nutzen?	17
2.3	Netzwerkzonen	19
	Internet	19
	Äußere DMZ (Frontend DMZ)	19
	Innere DMZ (Infrastructure DMZ)	19
	Hochsicherheitszone (Backend-LAN)	20
2.4	Web-Infrastruktur-Layout	20
	Netzwerkelemente	21
	Empfehlungen	23
3	Load Balancing	27
3.1	Load-Balancing-Verfahren	27
	Clientbasiertes Load Balancing	28
	DNS-basiertes Load Balancing	29
	Serverbasiertes Load Balancing	31
3.2	Load-Balancing-Algorithmen	34
3.3	Setup und Funktionalität des SAP Web Dispatchers	35
	Installation	36
	Initiale Anmeldung	38
	Änderung des Load-Balancing-Verfahrens	39
	Weitere Funktionen	41

4	Infrastruktursicherheit	43
4.1	Härtung der Infrastruktur	43
4.2	Firewalls	44
	Statische Paketfilter	44
	Dynamische Paketfilter	45
	Application Gateway	45
4.3	Konfigurationsanforderungen an Application Gateways	47
	Potenzielle Problemfelder	47
	Vermeidung von Problemen	48
4.4	Konfigurationsanforderungen an Anwendungen	49
	Host-Konfiguration in KMC	49
	Content Rewriting	51
4.5	Beispiel: Apache als Application Gateway für einen Application Server	52
	Basiskonfiguration Apache als Application Gateway	52
	Vervollständigung der grundlegenden Apache-Konfiguration und Verschlüsselung der Kommunikation	53
4.6	Beispiel: Apache als Application Gateway für mehrere Application Server	55
5	Anwendungssicherheit	57
5.1	Authentifizierung	57
	Form-based-Login/HTTP Basic Authentication	57
	X.509-Client-Zertifikate	57
	Headerbasierte Authentifizierung	58
	Integrated Windows Authentication	58
	Login-Module	58
	Container-based Authentication	59
	UME-based Authentication	60
5.2	Access Control Lists im SAP NetWeaver Portal	60
	Administrationsberechtigungen	61
	Endbenutzerberechtigungen	61
	Vererbung von Berechtigungen	62
5.3	Security Zones im Portal	63
	Ursprüngliches Security-Zone-Konzept	64
	Neues Security-Zone-Konzept	64
	Berechtigungsvergabe für Security Zones	65
	Security Zone Enforcement	65
5.4	Access Control Lists im Knowledge Management	66
5.5	Sicherheitsproblematiken im Web	67
5.6	Beispiel: Prüfung von Security Zones	68
5.7	Beispiel: Anwendung des Permission Viewers	69
6	Hochverfügbarkeit	71
6.1	Ungeplante Ausfallzeit	71
	Hochverfügbarkeitslösungen	72
	Hochverfügbarkeit für den SAP NetWeaver Application Server	72

Beispiel einer Hochverfügbarkeitsarchitektur für den SAP NetWeaver Application Server Java	73
Beispiel eines High-Availability-Setups für den SAP NetWeaver Application Server Dual Stack	75
6.2 Geplante Ausfallzeit	77
Nutzung eines Schattensystems zur Reduzierung von Planned Downtime	78
Nutzung von Virtualisierung zur Erstellung eines Schattensystems	81
Wartungsdurchführung in isolierter Netzwerkumgebung	82
Wartungsabschluss bzw. Rollback	82
Systemrollentausch (Switch)	83
7 Sizing und Lasttests	85
7.1 SAP-Sizing-Prozess	85
Prozessbeschreibung	86
Quick Sizer	87
Sizing-Varianten	88
Prozessunterstützung	89
7.2 Beispiel: Quick Sizer	89
Arbeiten mit dem Quick Sizer	90
7.3 Lasttests	93
Lasttestvarianten	93
Lasttestprozess	94
Lasttestsoftware	98
8 Performance	101
8.1 Client-Performance	101
8.2 Server- bzw. Anwendungsperformance	102
Einflussfaktoren auf die Anwendungsperformance	102
Datenbankperformance	102
Konfiguration der Laufzeitumgebung	103
Serverseitiges Caching	103
Performance der Anwendung	104
8.3 Netzwerkperformance	105
Einflussfaktoren auf die Netzwerkperformance	107
Paketverluste im Netzwerk	108
Netzwerklatenz	108
Bandbreite	111
8.4 Konfiguration des SAP NetWeaver Application Servers Java	111
Keep-Alive	111
Client-Caching	112
Komprimierung	114
8.5 Beispiel: Performanceanalyse	115
Beispielszenario	115
Messung der Netzwerkperformance	115
Messung der Client-Performance	116
Messung der Serverperformance	117

9 Weitere Zugriffsszenarien	119
9.1 Terminal-Server	119
Vorteile	120
Nachteile	120
Szenarien für den Einsatz	121
9.2 Virtual Private Networks	121
Vorteile	122
Nachteile	122
9.3 WAN-Beschleunigung	123
Web Proxy Caches	123
WAN-Beschleunigungshardware	124
Application und Content Delivery	125
SAP Application Delivery over WAN	125
9.4 Beispiel: Performanceverbesserung mit Squid	126
10 Zusammenfassung und Ausblick	129
Index	131

4 Infrastruktursicherheit

In diesem Kapitel wenden wir uns dem Thema Infrastruktursicherheit mit dem Schwerpunkt Application Gateways zu. Nach einem Überblick über generelle Sicherheitsfaktoren, die Sie in Ihrer Infrastruktur berücksichtigen sollten, geben wir Ihnen zunächst eine Einführung in das Thema Firewalls. Im Anschluss daran konzentrieren wir uns auf das Thema Application Gateways, insbesondere auf die Nutzung mit einem SAP NetWeaver Application Server. An den Theorie teil anschließend zeigen wir Ihnen, wie Sie den Apache-Webserver in einer SAP-Landschaft als Application Gateway einsetzen können.

4.1 Härtung der Infrastruktur

Aus Infrastrukturgesichtspunkten beeinflussen mehrere Faktoren die Sicherheit eines Gesamtsystems. Man spricht von *Härtung eines Systems* bzw. einer Infrastruktur, wenn diese Sicherheitsfaktoren adressiert werden. In diesem Abschnitt führen wir die wichtigsten Aspekte ein, die Ihnen bei der Definition eines sicheren Systems innerhalb eines Projektes helfen sollen.

► Firewall- und Netzwerksicherheit

Einen wichtigen Aspekt stellen Dienste dar, die innerhalb der Infrastruktur erreichbar sind, sogenannte *offene Ports*. Sofern diese Dienste notwendig sind und daher nicht abgeschaltet werden können, sollten Zugriffe durch strikte Firewall-Regeln mindestens eingeschränkt oder, wenn möglich, sogar verhindert werden.

Ein weiterer wichtiger Aspekt ist die Verschlüsselung des Netzwerkverkehrs. Dies betrifft zum einen die Kommunikation zwischen Client und Server und zum anderen die Kommunikation zwischen serverseitigen Komponenten. Sie sollten besonderes Augenmerk auf die allgemeinen Schutzziele (Integrität, Authentizität, Vertraulichkeit und Verbindlichkeit) legen. Verwen-

dete kryptografische Algorithmen sowie Schlüssellängen und Zertifikate sind dabei typische Auswahlkriterien in der Praxis. Hierzu liefern unter anderem das Bundesamt für Sicherheit in der Informationstechnik (<http://www.bsi.de>) und das Federal Bureau of Investigation (<http://www.fbi.gov>) entsprechende Empfehlungen.

► Betriebssystemsicherheit

Für die Betriebssysteme, auf denen Serverkomponenten installiert sind, gilt das Minimalprinzip, das heißt, nur die wirklich notwendigen Komponenten sowie Dienste sollten installiert sein. Jeder ungenutzte Dienst und jede ungenutzte Anwendung kann zu Sicherheitsproblemen führen. Für alle installierten Komponenten sollten immer der neueste Patch sowie die aktuellsten Security-Fixes eingespielt sein. Patches und Fixes sollten zudem nur von den Seiten des Herstellers bezogen werden.

Installierte Dienste und Anwendungen sollten im Kontext eines Benutzers mit eingeschränkten Betriebssystemrechten betrieben werden. Dies verhindert im Falle eines Einbruchs in das System, dass der Einbrecher sofort jegliche Betriebssystemberechtigung erlangt.

Auf der Seite des Betriebssystems sollte ein spezielles Augenmerk auf das Dateisystem gelegt werden. Vergeben Sie restriktive Dateisystem-Berechtigungen, damit nicht jeder direkt am System angemeldete Benutzer Zugriff auf schutzwürdige Informationen erhält. Zusätzlich muss sichergestellt werden, dass schützenswerte Daten im Dateisystem durch Verschlüsselungsmechanismen gesichert werden. Ein Beispiel hierfür ist der Secure Store des SAP NetWeaver Application Servers Java, in dem zum Beispiel das Datenbankzugriffspasswort abgelegt ist.

► Datenbanksicherheit

Wie beim Thema Betriebssystemsicherheit sollte auch auf Seiten der Datenbank sichergestellt sein, dass alle aktuellen Security-Fixes installiert sind. Zudem gilt, dass innerhalb der Datenbank entsprechende Berechtigungen beispielsweise auf Schemaebene existieren, damit nicht ein beliebiger Datenbankbenutzer Zugriff auf Daten aus einem geschützten Bereich erhält.

► Directory-Server-Sicherheit

Bei der Sicherheit eines eventuell eingesetzten Directory-Servers ist wiederum darauf zu achten, dass über entsprechende Berechtigungen schützenswürdige Bereiche vor Benutzern verborgen werden und das System auf dem aktuellen Patch-Stand gehalten wird.

► Security Audit Logging

Aufgabe des Security Audit Loggings ist die Dokumentation sicherheitskritischer Änderungen, zum Beispiel von Berechtigungsänderungen. Das Security Audit Logging stellt somit einen wichtigen Faktor für die Nachvollziehbarkeit von Änderungen in einem System dar. Es ist wichtig, dass Sie sich mit der entsprechenden Funktionalität der in Ihrem Unternehmen eingesetzten Software auseinandersetzen. Für den AS Java finden Sie zum Beispiel Details in der SAP NetWeaver-Dokumentation im SAP Help Portal auf <http://help.sap.com> mit dem Suchbegriff »Security Audit Log of the AS Java«.

► Anwendungssicherheit

In Kapitel 5, »Anwendungssicherheit«, gehen wir auf diesen Aspekt im Detail ein. Wichtige Themen in diesem Bereich sind unter anderem Anwendungsberechtigungen und für den Systembetrieb notwendige Anwendungsdienste. Welche Dienste für welche Szenarien deaktiviert werden können, beschreibt SAP-Hinweis 871394.

► Physische Sicherheit

Ergänzend zu den bisher genannten Punkten ist die physische Sicherheit der Server zu betrachten. Der physische Zugriff auf die Server im Rechenzentrum darf nur einem kleinen Kreis autorisierter Personen möglich sein. Außerdem sollten Backup-/Restore-Prozeduren definiert sein, um die Datensicherheit des Systems im Falle eines Hardwareproblems etc. sicherzustellen. Hierzu gehört auch die sichere Verwahrung der Backup-Medien.

In diesem Heft adressieren wir hauptsächlich die Themen Anwendungs- sowie Netzwerksicherheit. Sofern Sie weitere Details zu den thematisierten Bereichen benötigen, empfehlen wir Ihnen die SAP Security Guides der einzelnen Produkte, die hierzu zahlreiche Informationen enthalten. Sie finden die Security Guides im SAP Help Portal (<http://help.sap.com>) in den einzelnen Anwendungsbereichen.

4.2 Firewalls

Bei Firewalls handelt es sich um Sicherheitskomponenten, die in einem Netzwerk eingesetzt werden. Sie dienen dazu, den Verkehr zwischen bestimmten Netzwerksegmenten abzusichern bzw. zu reglementieren. Wie Sie in Kapitel 2, »Web-Infrastrukturen«, gesehen haben, empfehlen wir den Einsatz einer Firewall zwischen allen Netzwerksegmenten:

- zwischen Internet und äußerer DMZ
- zwischen äußerer DMZ und innerer DMZ
- zwischen innerer DMZ und Hochsicherheitszone
- zwischen Intranet und Hochsicherheitszone
- zwischen Intranet und DMZ

Firewalls sind von einfachen, sogenannten statischen Paketfiltern über ausgereifte dynamische Paketfilter bis hin zu Application Gateways oder Application Layer Firewalls weiterentwickelt worden. Auf die Einzelheiten dieser Firewall-Typen gehen wir im Folgenden näher ein.

Statische Paketfilter

Statische Paketfilter werden auch als Firewalls der ersten Generation bezeichnet. Diese entscheiden anhand der Informationen im IP-Header, ob ein Paket weitergeleitet wird oder nicht. Innerhalb dieses Headers stehen Informationen wie Quelladresse, Zieladresse, also auch die Richtung der Kommunikation, sowie der Serviceport zur Verfügung. Man spricht von statischen Paketfiltern, da die »Lücken« in der Firewall permanent bestehen bleiben und nicht nach Bedarf geöffnet bzw. geschlossen werden.

Das Verfahren ist sehr verbreitet, da es als klassische Router-Lösung in jedem Router verfügbar ist. In Abbildung 4.1 sehen Sie eine Firewall, die den Zugriff über

Port 443 (normalerweise für HTTPS genutzt) zulässt. Alle Anfragen an andere Ports sind geblockt.

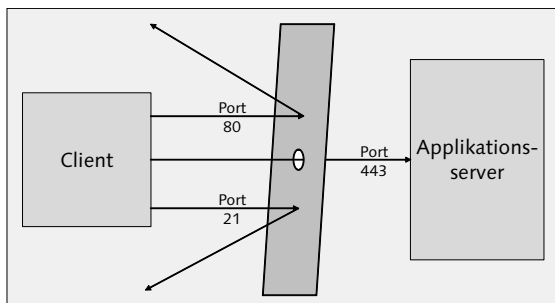


Abbildung 4.1 Statischer Paketfilter

Das Verfahren bietet folgende Vorteile:

- ▶ Es bietet einen hohen Datendurchsatz.
- ▶ Es ist transparent für die Anwendung, eine Anpassung aufseiten der Anwendung ist daher nicht nötig.
- ▶ Es ist sehr kostengünstig.
- ▶ Das Setup ist einfach.

Neben diesen Vorteilen ergeben sich allerdings auch gravierende Nachteile:

- ▶ Es bestehen direkte Kommunikationsverbindungen zwischen Quelle und Ziel.
- ▶ Das Verfahren ist anfällig für IP-Spoofing-Attacken.
- ▶ Das Regelwerk kann kompliziert und dadurch fehleranfällig werden.
- ▶ Eine Benutzerauthentifizierung ist nicht möglich.
- ▶ Angriffe auf Anwendungsebene können nicht erkannt werden.

Dynamische Paketfilter

Bei *dynamischen Paketfiltern* (auch *Stateful Packet Filter* bzw. *Stateful Inspection*) handelt es sich um die klassische Firewall-Lösung, die einige Schwächen von statischen Paketfiltern beseitigt. Man spricht daher von Firewalls der zweiten Generation.

Im Gegensatz zu statischen Paketfiltern werden bei dynamischen Paketfiltern die Verbindungskanäle nicht offen gehalten, sondern immer wieder geschlossen, da neben den reinen IP-Informationen zusätzlich Protokollinformationen ausgewertet werden. Dadurch ist es möglich, Informationen zum Status einer Verbindung zu erhalten und die Filterregeln an Protokollspezifikationen anzupassen. Dynamische Paketfilter können somit

Gefahren erkennen, die beim reinen statischen Paketfilter nicht erkannt werden, indem die IP-Pakete nicht einzeln, sondern im Protokollkontext betrachtet werden.

Abbildung 4.2 zeigt, dass neben der reinen Portinformation außerdem überprüft wird, ob es sich bei dem Protokoll um HTTP handelt. Falls nicht, wird die Verbindung untersagt.

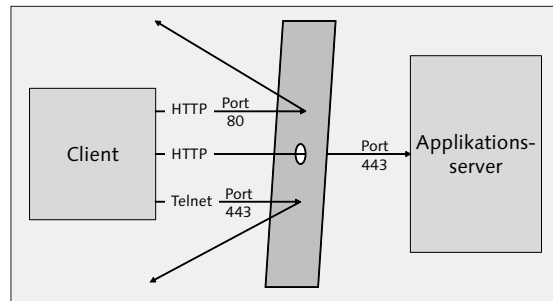


Abbildung 4.2 Dynamischer Paketfilter

Dynamische Paketfilter bieten vielfältige Vorteile:

- ▶ Sie erlauben einen hohen Datendurchsatz.
- ▶ Lücken sind nur zeitweise vorhanden, da die Verbindungskanäle immer wieder geschlossen werden.
- ▶ Sie bieten die Unterstützung fast aller Dienste.

Neben den genannten Vorteilen ergeben sich allerdings ebenfalls wieder einige Nachteile:

- ▶ Es bestehen direkte Kommunikationsverbindungen zwischen Quelle und Ziel.
- ▶ Eine Benutzerauthentifizierung ist nicht möglich.
- ▶ Angriffe auf Anwendungsebene können nicht erkannt werden.

Application Gateway

Application Gateways oder auch *Application Layer Firewalls* prüfen den Inhalt der übertragenen Daten auf Anwendungsebene. Ein Application Gateway kann beispielsweise Informationen innerhalb des HTTP-Datenstroms auswerten und auf Basis dieser Informationen entscheiden, ob eine Anfrage erlaubt oder blockiert wird.

Die meisten Application Gateways beinhalten integrierte Proxys. Dies bedeutet, dass es für den Empfänger der Informationen so aussieht, als ob das Application Gateway den Absender der Informationen darstellt. Auf der anderen Seite stellt das Application Gateway für den

Application Server den Client dar. Somit agiert das Application Gateway gewissermaßen als Mittelsmann.

Mittelsmann Application Gateway

Man könnte das Application Gateway mit einem Anwalt vergleichen, der zwischen zwei Parteien vermittelt, ohne dass sich die zwei Parteien direkt miteinander unterhalten. Die Parteien müssen sich darüber hinaus gegenseitig nicht kennen bzw. nicht von ihrer Existenz wissen.

Für die Kommunikation werden zwei separate TCP-Verbindungen benötigt, wie in Abbildung 4.3 dargestellt ist:

- Client zu Application Gateway
- Application Gateway zu Applikationsserver

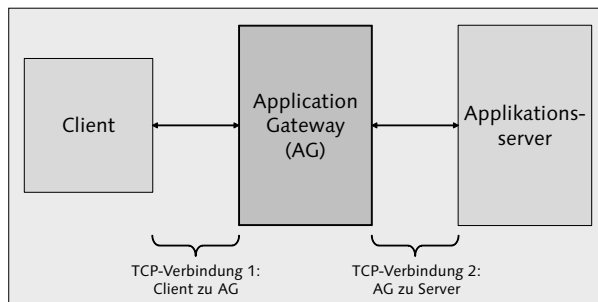


Abbildung 4.3 Application Gateway

Zwischen Client und Server besteht somit keine durchgehende TCP-Verbindung.

Darüber hinaus ermöglicht die Proxy-Funktionalität, dass bestimmte Informationen auf dem Application Gateway zwischengespeichert/gecacht werden. In diesem Zusammenhang wird auch immer wieder von *Reverse-Proxy-Servern* gesprochen. Reverse-Proxy-Server ordnet man somit in die Kategorie der Application Gateways ein. Ein Application Gateway ist, genau genommen, mehr als ein Reverse-Proxy, da es meistens mit zusätzlichen Firewall-Funktionalitäten ausgestattet ist.

Der Ausdruck Reverse-Proxy leitet sich davon ab, dass ein solcher Proxy genau umgekehrt zu einem normalen Webproxy funktioniert. Ein Webproxy stellt für einen bzw. mehrere Clients innerhalb eines geschützten Netzwerks eine Verbindung zu vielen Webservern außerhalb dieses Netzwerks dar – eine direkte Kommunikation mit dem Application Server ist nicht möglich. Die Verwen-

dung eines Reverse-Proxys beruht auf der Idee, dass der Server in einem geschützten Netzwerk und die Clients in einem eventuell ungeschützten oder nicht vertrauenswürdigen Netzwerk stehen.

Application Gateways bieten folgende Vorteile:

- Sie ermöglichen die Analyse des Datenverkehrs auf Anwendungsebene.
- Eine direkte Kommunikationsverbindung zwischen Quelle und Ziel existiert nicht.
- Sie bieten die Unterstützung zusätzlicher Authentifizierung (sofern gewünscht), bevor der Request den Application Server erreicht.
- Umfassendes Logging ist möglich, sofern der Datenstrom im Klartext vorliegt.

Obwohl es sich bei Application Gateways um die sichersten erhältlichen Firewalls handelt, bestehen auch hier einige Nachteile:

- Sie sind langsamer als Paketfilter.
- Eventuell müssen Konfigurationsanpassungen der Anwendungen vorgenommen werden.
- Sie unterstützen nicht jede Art von Anwendung/Dienst (abhängig vom jeweiligen Produkt).
- Man muss häufig eine komplexe Konfiguration in Kauf nehmen, was zu Fehlern führen kann.

Welches Application-Gateway-Produkt?

Abhängig von Ihren Sicherheitsanforderungen ergibt sich die Möglichkeit, aus verschiedenen Application-Gateway-Produkten bzw. Produkten mit Application-Gateway-Funktionalitäten zu wählen. Nachfolgend finden Sie eine Auswahl:

► SAP Web Dispatcher

Der SAP Web Dispatcher ist mit grundlegenden Application-Gateway-Funktionalitäten ausgestattet (siehe auch Kapitel 3, »Load Balancing«). In diesem Heft konzentrieren wir uns allerdings auf seine Load-Balancing-Funktionalitäten, um eine möglichst heterogene und generische Systeminfrastruktur vorzustellen. Anstelle des Apache-Servers, den wir in unserer Beispielkonfiguration benutzen, kann eventuell auch der SAP Web Dispatcher eingesetzt werden.

► Apache-Webserver

Der Apache-Webserver bietet ebenfalls Application-Gateway-Funktionalität. So lassen sich zum Beispiel auf der Basis von regulären Ausdrücken komplexe Weiterleitungsregeln konfigurieren. Alternativ kann durch eigene Module der Datenstrom beliebig angepasst werden. Daher ist Apache als Webserver und Application Gateway häufig im Einsatz.

Da es sich bei Apache um ein Open-Source-Produkt handelt, stellt sich allerdings eventuell die Frage nach dem Support für die Nutzung in einem Produktivszenario.

► Spezielles Sicherheitsprodukt

Auf dem Markt sind viele spezialisierte Application-Gateway-Produkte verfügbar, die über vielfältige Konfigurationsoptionen verfügen. Oft handelt es sich um kombinierte Software- und Hardwarekomponenten, die in Ihr Netzwerk als Appliance integriert werden können. Für ein System mit höchsten Sicherheitsanforderungen wird ein solches Produkt wohl in die engere Auswahl kommen.

Nicht zu vernachlässigen ist allerdings der Kosten-Nutzen-Aspekt. Dieser sollte an den individuellen Schutzziele ausgerichtet sein und spricht eventuell gegen ein solches Produkt.

► Probleme mit Servernamen

Aufgrund unterschiedlicher DNS-Auflösung ruft ein Client den Application Server in der Regel mit einem anderen Hostnamen bzw. *Full Qualified Domain Name* (FQDN) auf als das Application Gateway. Dies kann dazu führen, dass ein Server auf eine Anfrage über das Application Gateway eine URL als Antwort sendet, die der Client nicht erreichen kann; damit tritt eine Fehlersituation auf.

Webanwendungen sollten immer relative URLs zur Referenzierung anderer Ressourcen benutzen, also URLs, die keine Serverinformationen beinhalten, zum Beispiel `/irj/servlet/prt/portal/prtroot/test.app`. In manchen Situationen sind absolute URLs, wie beispielsweise `http://portal.domain.com/test`, nicht zu vermeiden. Dies ist unter anderem der Fall, wenn eine Rückreferenz an eine Anwendung geschickt werden muss; zum Beispiel im Fall, dass das Portal einer Anwendung den Link auf die Portal-Stylesheets übergibt, sodass aus dieser Anwendung heraus auf die Portal-Stylesheets zugegriffen werden kann. Dies ist erwünscht, um ein einheitliches Aussehen der Applikation zu erreichen. Des Weiteren muss zum Beispiel eine absolute URL in E-Mail-Benachrichtigungen verschickt werden.

Die Problematik mit Servernamen kann auch im Zusammenhang mit serverseitigen Redirects auftreten, das heißt, wenn ein Server eine Client-Anfrage auf eine andere Ressource bzw. ein anderes Verzeichnis umleitet und dabei ein absolutes Adressierungsschema verwendet.

► Protokollprobleme

Eine gängige Praxis ist, dass auf einem Application Gateway eine verschlüsselte Verbindung terminiert wird. Aus Sicherheitsgesichtspunkten kann dies akzeptabel sein, sofern die Kommunikation vom Application Gateway zum Application Server in einer geschützten Netzwerkumgebung stattfindet.

Der Vorteil einer solchen Konfiguration ist die Entlastung des Application Servers. Die für eine SSL-Verbindung notwendige Ver- und Entschlüsselung findet auf dem Application Gateway statt, dies wird als *Offloading* bezeichnet. Zusätzlich wird der Konfigurationsaufwand reduziert, da keine Zertifikate in die Application Server eingespielt werden müssen.

4.3 Konfigurationsanforderungen an Application Gateways

Als einen der Nachteile eines Application Gateways haben wir die möglicherweise fehlende Transparenz auf Anwendungsebene genannt. Der Grund ist, dass ein Application Gateway die Kommunikation auf Anwendungsebene unterbricht – im Falle einer Webanwendung wird die Kommunikation auf HTTP-Ebene unterbrochen.

Potenzielle Problemfelder

Aus den zwei voneinander unabhängigen Kommunikationsverbindungen können mehrere Probleme resultieren, die nachfolgend diskutiert werden sollen.

Der Wechsel des Protokolls von HTTPS zu HTTP kann aber wiederum zu Problemen mit absoluten URLs auf dem Application Server führen, da eine HTTP-Anfrage an den Server gestellt wird und für den Server zunächst nicht ersichtlich ist, dass der Client eine SSL-Anfrage gestellt hat.

Aus Performancegründen und zur Gewährleistung einer fehlerfreien Kommunikation mit dem AS Java sollten Sie darauf achten, dass Ihr Application Gateway HTTP 1.1 unterstützt.

► **Cookie-Probleme**

Ein weiteres denkbares Problemfeld stellen Cookies dar. Cookies werden über den HTTP-Header `Set-Cookie` ausgegeben. Hier ist es möglich, dass ein Server ein Cookie für eine bestimmte Domäne ausstellt, zum Beispiel *company.com*. Dies geschieht durch folgenden Eintrag im `Set-Cookie`-Header:

```
domain=company.com
```

Wird nun das Application Gateway über die Domäne *company.net* angesprochen, akzeptiert der Browser die Domäne *company.com* nicht, da dies die Cookie-Spezifikation RFC 2109 verbietet (siehe <http://rfc.dotsrc.org/rfc/rfc2109.html>). Der Browser wird daher den Hostnamen des Servers als gültige Cookie-Domäne wählen. Somit erhält der Client ein Cookie, das nur für *server.company.net* gültig ist und nicht, wie gewünscht, für die Domäne *company.com*. Dies führt gewöhnlich zu Folgeproblemen.

► **Anmeldeinformationen**

Für den Fall, dass auf dem Application Gateway aus Sicherheitsgründen eine Anmeldung durchgeführt werden soll, müssen Sie berücksichtigen, dass der Application Server diese Informationen erhält. Dies gilt selbstverständlich nur für den Fall, dass keine zweistufige Anmeldung gewünscht ist. Bei HTTP-Parametern stellt dies kein Problem dar. Schwierigkeiten entstehen aber in aller Regel, wenn komplexere Anmeldemechanismen wie beispielsweise X.509-Zertifikate zum Einsatz kommen. Hier müssen Sie über geeignete Mechanismen im Application Gateway sicherstellen, dass die Anmeldeinformationen zum Beispiel in einem HTTP-Header an den Application Server weitergeleitet werden.

Auch wenn diese Punkte kritisch klingen mögen, so sind sie in der Regel recht einfach zu vermeiden. Gleichwohl ist es aus unserer Erfahrung wichtig, dass Sie sich potenzielle Probleme bewusst machen, um eventuell auftretende Komplikationen korrekt einschätzen zu können. Die notwendigen Anforderungen, die ein Application Gateway im SAP-Umfeld erfüllen muss, sind in SAP-Hinweis 833960 dokumentiert.

Vermeidung von Problemen

Um die benannten Probleme zu vermeiden, muss der Application Server Kenntnis darüber erlangen, wie der Client das Application Gateway anspricht. Dies bezieht sich auf den Hostnamen, den Port und auf das verwendete Protokoll.

► **Host und Port**

Hostname und Port sind über den HTTP-Header `Host` (sogenannter *Host-Header*) verfügbar. Damit der Application Server korrekt auf die Anfrage reagieren kann, muss ihn dieser HTTP-Header erreichen. Hierfür ist das Application Gateway verantwortlich. Der SAP NetWeaver AS liefert dann seinerseits den lokalen Anwendungen die Information aus dem Host-Header, in Java erfolgt dies über das `HttpServletRequest`-Objekt.

► **Protokoll**

Da das Protokoll nicht im Host-Header erscheint und sich auf dem Application Gateway, bedingt durch die SSL-Terminierung, verändert, das heißt, ein Protokoll-Switch stattfindet, muss das Application Gateway diese Information ebenfalls dem Application Server mitteilen. Analog zu den Informationen aus dem Host-Header bezieht der Application Server diese Information aus einem HTTP-Header, der durch das Application Gateway gefüllt werden muss. Das Protokoll wird anschließend transparent an die Anwendung weitergegeben.

Der Name des HTTP-Headers ist im AS Java in den Optionen des HTTP-Provider-Service konfigurierbar. Standardmäßig ist der Name des HTTP-Headers `ClientProtocol`. Um den Namen neu zu konfigurieren, gehen Sie bitte wie im folgenden Kasten, »Konfiguration des Client-Protokoll-Headers (AS Java)«, beschrieben vor.

Konfiguration des Client-Protokoll-Headers (AS Java)

Für den Fall, dass Ihr Application Gateway für die Weitergabe der Protokollinformation nicht den HTTP-Header `ClientProtocol`, sondern einen anderen Namen verwenden soll, gehen Sie folgendermaßen vor:

1. Starten Sie den Visual Administrator im Verzeichnis `/usr/sap/<SID>/<Instanz>/j2ee/admin` mit dem Kommando `go` bzw. `go.bat`, und melden Sie sich mit einem **Administrator**-Account an.
2. Navigieren Sie zur Einstellung des Serverknotens über **Cluster • Instanz • Server <x>**.
3. Wählen Sie den **HTTP-Provider-Service (Services • HTTP Provider)**.
4. Auf dem Reiter **Properties** ändern Sie nun den Wert des Parameters `ProtocolHeaderName` nach Ihren Bedürfnissen (siehe Abbildung 4.4).
5. Speichern Sie Ihre Einstellungen.

4.4 Konfigurationsanforderungen an Anwendungen

Möglich ist, dass Anwendungen die Informationen, die der Application Server aus dem Host-Header ausliest und zur Verfügung stellt, nicht nutzen oder auch nicht nutzen können. Anwendungen können zum Beispiel die Informationen nicht verwenden, sobald eine asynchrone

Kommunikation mit dem Benutzer stattfindet, also kein aktueller Benutzer-Request vorhanden ist, aus dem die Informationen bezogen werden könnten.

Host-Konfiguration in KMC

Ein Beispiel für eine asynchrone Kommunikation findet man im Bereich Knowledge Management und Collaboration (KMC), wenn Benutzer eine Benachrichtigung erhalten, dass sich zum Beispiel ein bestimmtes Dokument geändert hat (*Subscription Notifications*). Zu diesem Zeitpunkt liegt keine aktuelle Benutzeranfrage vor. Somit muss die in der Benachrichtigung verschickte URL aus einem Konfigurationsparameter bezogen werden.

Diese URL konfigurieren Sie wie folgt:

1. Rufen Sie das SAP NetWeaver Portal auf, und melden Sie sich mit dem **Administrator**-Account an, oder benutzen Sie alternativ einen Benutzer, der über die Rolle `pcd:portal_content/administrator/super_admin/super_admin_role` verfügt.
2. Navigieren Sie über **System Administration • System Configuration • Knowledge Management • Content Management** zur Content-Management-Administration.
3. Über **Mode • Advanced** wechseln Sie in die erweiterte Konfiguration.
4. Navigieren Sie zu **Global Services • URL Generator Service**.

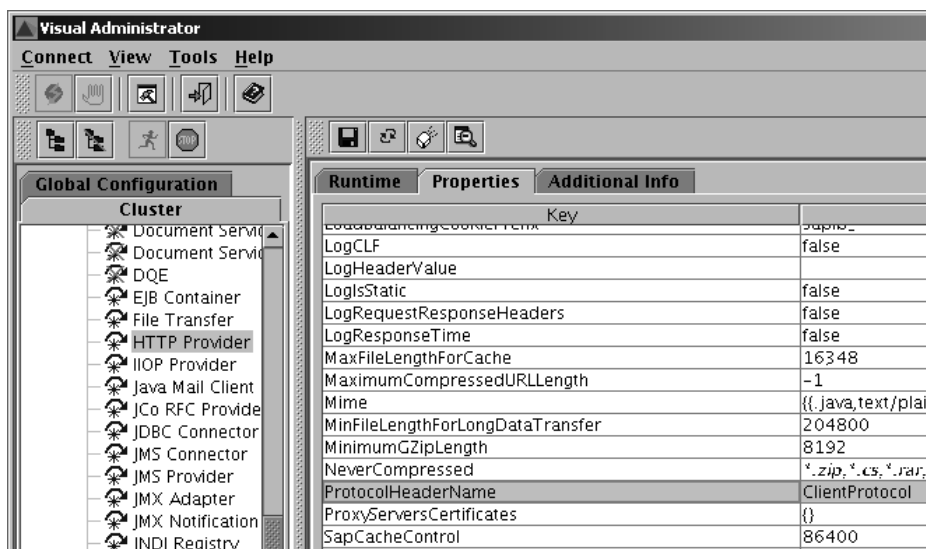


Abbildung 4.4 Änderung des Protokoll-Header-Namens

5. An dieser Stelle pflegen Sie nun Protokoll, Hostnamen sowie Port im Parameter **Host** mit der URL ein, die Ihre Endbenutzer für den Aufruf des Systems über das Application Gateway benutzen (siehe Abbildung 4.5). Der Eintrag **Alternative Host** wird für die interne Kommunikation der Such- und Klassifizierungsmaschine zu KMC benutzt und enthält daher meistens eine URL, die eine direkte Kommunikation der Such- und Klassifizierungsmaschine (TREX) mit den Portal-Servern bzw. einem Portal-Server ermöglicht.



Abbildung 4.5 URL-Generator-Konfiguration

Falls das System sowohl für interne Benutzer als auch externe Benutzer im Einsatz ist, muss die im **Host**-Parameter gepflegte URL sowohl von internen als auch externen Benutzern erreichbar sein. Es ist nicht möglich, eine Unterscheidung dieser Benutzergruppen festzulegen.

Ein identischer Hostname für externe sowie interne Benutzer muss allerdings nicht bedeuten, dass die Zugriffspfade der Benutzer identisch sein müssen, dass demnach zum Beispiel ein Benutzer im firmeninternen Netzwerk nur über das Internet auf die Anwendung zugreifen kann. Eine Unterscheidung ist durch eine unterschiedliche DNS-Konfiguration möglich:

- ▶ Der interne DNS-Server löst den Hostnamen bzw. FQDN in eine IP-Adresse auf, die nur aus dem internen Netzwerk zu erreichen ist.
- ▶ Der externe DNS-Server löst den Hostnamen bzw. FQDN in eine öffentliche IP-Adresse auf, auf die aus dem Internet zugegriffen werden kann.

Ähnliches gilt auch für Anwendungen, die per URL in das Portal eingebunden werden. Hier sollten Sie ebenfalls sicherstellen, dass der Hostname sowohl intern als auch extern erkannt und wiederum vielleicht in verschiedene IP-Adressen aufgelöst wird. Des Weiteren ist es wichtig zu erwähnen, dass das Portal bei einer URL-Integration nicht als Proxy für das eingebundene System dient.

SAP NetWeaver Portal als Content-Proxy?

Häufig wird fälschlicherweise angenommen, dass das SAP NetWeaver Portal als eine Art Content-Proxy für Anwendungen dient, die per URL-Integration eingebunden sind. Dies ist nicht der Fall. Content-Proxy würde in diesem Fall bedeuten, dass der Client nicht direkt mit der Anwendung kommuniziert, sondern nur mit dem Portal, das die Anfragen weiterleitet.

Innerhalb des Portals gibt es prinzipiell zwei Standardmöglichkeiten, um Content in Form einer URL-Integration einzubinden:

▶ Application-Integrator-iView

Der **Application-Integrator-iView** ist in der Lage, in Abhängigkeit eines Systemobjekts, das Zugriffsinformationen festlegt, sowie in Abhängigkeit von bestimmten Benutzereinstellungen, den Benutzer per Redirect an eine Anwendung zu senden, die dann benutzerspezifisch reagiert. Redirect bedeutet, dass der Client direkten Zugriff auf den Application Server haben muss – natürlich kann aber auch hierbei wieder ein Application Gateway verwendet werden.

Die einzige Ausnahme bilden an dieser Stelle iViews für SAP BW 3.x, für die **iView-Caching** über die iView-Propertys **Cache Level** sowie **Cache Validity Period** aktiviert ist. Nur für diese Art von iViews ist kein Zugriff auf das Backend-System notwendig.

Details hierzu finden Sie unter dem Stichwort »BEx Web Application or Query as iView in the Portal« in der SAP NetWeaver-Dokumentation im SAP Help Portal.

▶ URL-iView

URL-iViews gibt es in zwei Ausprägungen, in einer clientseitigen sowie in einer serverseitigen Variante. Dies wird über den iView-Parameter **Fetch Mode** in der Parameterkategorie **Advanced** gesteuert (siehe Abbildung 4.6).

Der Unterschied der beiden Ausprägungen liegt in der Art und Weise, wie der initiale HTML-Code der Seite an den Client gesendet wird. Bei der serverseitigen Variante bezieht der Portal-Server den HTML-Code von der einzubindenden Anwendung und sendet ihn dann an den Client. Damit ist es

möglich, den initialen HTML-Code auf Portal-Seite zu cachen. Bei der clientseitigen Variante bezieht der Client über einen vom Portal-Server initiierten Redirect den HTML-Code direkt von der Anwendung oder indirekt über ein Application Gateway.

Abbildung 4.6 iView Fetch Mode

Wichtig zu erwähnen ist, dass jeder Folge-Request in beiden Fällen immer über die Anwendungs-URL erfolgt. Diese Folge-Requests sind zum Beispiel für Stylesheets, Bilder etc. erforderlich.

Abbildung 4.7 zeigt den Request-Flow bei einem serverseitigen URL-iView. Hier sehen Sie, wie zunächst der Client beim Server den iView anfragt (1). Sofern der iView noch nicht auf dem Server gecached ist, stellt der Server eine Anfrage (2a) und empfängt die Antwort (2b). Den empfangenen HTML-Code sendet der Server dann an den Client (3). Die weitere Kommunikation findet direkt zwischen Client und Anwendung statt (...).

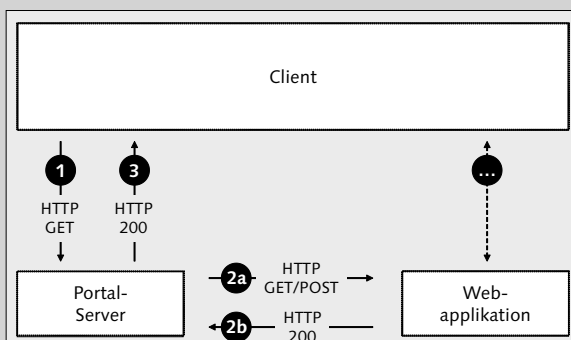


Abbildung 4.7 Serverseitiger URL-iView

Content Rewriting

Manche Anwendungen generieren trotz fehlender Notwendigkeit absolute URLs auf der Basis von Konfigurationseinträgen. Falls Sie Einfluss auf die Anwendungsentwicklung haben, sollten Sie diesen Punkt entsprechend adressieren.

Für den Fall, dass es zeitnah nicht gelingt, die Anwendung zu beeinflussen, besteht noch die Möglichkeit des sogenannten *Content Rewritings* auf dem Application Gateway. Dies bedeutet, dass das Application Gateway nicht nur die HTTP-Header-Informationen, sondern auch statische Content-Informationen, wie beispielsweise den HTML-Code, entsprechend den Anforderungen umschreibt.

Dieses Verfahren wird von einigen Application Gateways angeboten, hat aber mehrere Nachteile, weshalb Sie dies nur in Ausnahmesituationen einsetzen sollten:

► Fehleranfälligkeit

Ein großes Problem stellt die Fehleranfälligkeit von Content Rewriting dar. Zum einen muss jede mögliche Variante bedacht werden, wie die absolute URL generiert werden kann, zum Beispiel auch innerhalb von JavaScript auf Client-Seite, und zum anderen bedeutet jede neue Version der einzubindenden Anwendung potenziellen Nachbesserungsbedarf bei den Rewrite-Regeln.

► Erhöhter Ressourcenbedarf und Geschwindigkeits-einbußen

Der komplette Content muss analysiert (geparst) und eventuell umgeschrieben werden, was aufgrund aufwendiger String-Operationen zu erhöhtem Bedarf an Systemressourcen, vorrangig CPU und Speicher, auf dem Application Gateway führt. Zudem bedeuten diese String-Operationen je nach Regelwerk einen gewissen Geschwindigkeitsnachteil bei der Beantwortung des Requests.

► Anwendungswissen erforderlich

Wie beim Aspekt Fehleranfälligkeit angedeutet, können die Ersetzungsoperationen je nach Anwendung recht komplex werden. Um hier eine wirklich ganzheitliche Sicht zu erhalten und nicht eventuell bestimmte Szenarien zu übersehen, ist es wichtig, die Anwendung sehr gut zu kennen.

► Testaufwand

Um sicherzustellen, dass nicht bestimmte Szenarien übersehen wurden, ist ein kontinuierlicher Testaufwand erforderlich, da sich die eingebundene Anwendung je nach Version unterschiedlich verhalten kann. Auf dem Apache-Server existiert für das Umschreiben von Content zum Beispiel das Modul `mod_ext_filter`, mit dem Sie eine Filterfunktionalität umsetzen können. Die Apache-Dokumentation empfiehlt allerdings, dieses Modul nur zu Testzwecken zu nutzen. Daher gibt es von verschiedenen Anbietern jeweils eigene Filtermodule, die für diesen Zweck in den Apache-Server geladen werden können.

4.5 Beispiel: Apache als Application Gateway für einen Application Server

Das folgende Beispiel stellt Ihnen eine Apache-Konfiguration vor, bei der der Apache-Server als Application Gateway agiert. Zunächst beschreiben wir eine einfache Konfiguration mit einer unverschlüsselten Verbindung zwischen Client und Application Gateway.

Sicherheitshinweis

Die vorgestellte Apache-Konfiguration soll ausschließlich dazu dienen, die grundlegenden Ideen zu vermitteln, sie entspricht nicht den Sicherheitsanforderungen eines Produktivsystems. Um die Konfiguration für Produktivsysteme nutzen zu können, müssen Sie den Apache-Server entsprechend den aktuellen Sicherheitsempfehlungen konfigurieren. Die hier beschriebene Konfiguration ist eine Ausgangsbasis und muss entsprechend Ihren Sicherheitsanforderungen weiter verfeinert werden.

Da dies häufig aus Sicherheitsgründen nicht für den Einsatz in einer produktiven Umgebung ausreicht, zeigen wir Ihnen im zweiten Teil dieses Abschnitts noch eine Konfiguration, bei der eine verschlüsselte Verbindung (HTTPS) zwischen Client und Application Gateway besteht. Die verschlüsselte Verbindung wird dazu auf dem Apache-Server terminiert. Hierbei können zusätzlich sogenannte *Cryptographic Hardware Accelerators* zum Einsatz kommen, um ein Offloading des SSL Processing Overheads zu erreichen. Der Apache-Server kommuniziert dann ver-

schlüsselt mit dem Client und unverschlüsselt über HTTP mit dem Application Server. Wie oben beschrieben, setzt diese Konfigurationsoption voraus, dass sich der Apache-Server sowie der Application Server in demselben geschützten Netzwerksegment befinden.

Basiskonfiguration Apache als Application Gateway

Für unser Beispiel verwenden wir die Version 2.2.4 des Apache-Webserver. Folgende Apache-Module sind für die Durchführung der Beispielkonfiguration erforderlich:

- `mod_rewrite`
- `mod_proxy`
- `mod_proxy_http`
- `mod_headers`
- `mod_ssl`

1. Sofern die Module nicht in den Apache-Server hinein kompiliert sind, müssen sie separat geladen werden. Dies geschieht in der zentralen Konfigurationsdatei des Apache-Servers:

```
<ApacheVerzeichnis>/conf/httpd.conf.
```

2. Sie müssen folgende Einträge in der Konfigurationsdatei pflegen, damit die Module geladen werden:

```
LoadModule rewrite_module modules/
    mod_rewrite.so
LoadModule proxy_module modules/
    mod_proxy.so
LoadModule proxy_http_module modules/
    mod_proxy_http.so
LoadModule headers_module modules/
    mod_headers.so
LoadModule ssl_module modules/
    mod_ssl.so
```

3. Nachdem die notwendigen Module geladen sind, müssen sie konfiguriert werden. Dies geschieht wiederum in der Apache-Konfigurationsdatei.

4. Für die Weiterleitung des Requests vom Apache-Server an Ihren Portal-Server bzw. an Ihren Application Server müssen Sie zunächst die Rewrite-Engine aktivieren. Die Rewrite-Engine sorgt dafür, dass Requests des Clients korrekt an den Application Server weitergeschickt bzw. umgeschrieben werden. Dies geschieht durch folgende Anweisung in der Apache-Konfigurationsdatei:

```
RewriteEngine On
```

5. Um den Sicherheitsanforderungen gerecht zu werden und außerdem für die Tests einen guten Überblick über die Weiterleitung zu erhalten, aktivieren Sie in der Apache-Konfiguration außerdem das Logging der Rewrite-Engine. Hierzu definieren Sie den Pfad zur Log-Datei sowie den Log-Level. Mögliche Werte für den Log-Level sind 0 bis 9, wobei 0 kein Logging bedeutet und 9 das maximal mögliche Logging. Ein Log-Level größer 2 ist allerdings für den Produktivbetrieb aus Performancegründen nicht empfohlen. Folgende Parameter bieten sich für eine Testkonfiguration an:

```
RewriteLog logs/rewrite.log
RewriteLogLevel 3
```

6. Bezüglich des Rewritings fehlt nun noch die Information, welche Requests wohin weitergeleitet werden. Dies geschieht über folgende Konfiguration:

```
RewriteRule ^/irj/(.*)
    http://<server>:<port>/irj/$1?%
    {QUERY_STRING} [P,L]
RewriteRule ^/logon/(.*)
    http://<server>:<port>/logon/$1?%
    {QUERY_STRING} [P,L]
RewriteRule ^/webdynpro/(.*)
    http://<server>:<port>/webdynpro/$1?%
    {QUERY_STRING} [P,L]
```

Jede der drei Konfigurationszeilen ist für die Weiterleitung eines bestimmten Pfads zuständig:

- `/irj`
Pfad für das Portal
- `/logon`
Pfad für die Standard-Logon-Grafiken sowie das Stylesheet. Diese Grafiken können über UME-Konfigurationsparameter geändert werden (`ume.logon.branding_image`, `ume.logon.branding_style`, `ume.logon.branding_text`), wodurch diese Weiterleitung vermieden werden kann.
- `/webdynpro`
Einstiegspfad für Web-Dynpro-Anwendungen

Diese drei Weiterleitungen sind für ein rudimentäres Portalszenario inklusive der Nutzung von Web-Dynpro-Anwendungen auf dem AS Java ausreichend. Ersetzen Sie die Einträge `<server>` sowie `<port>` durch die Informationen Ihres Portals bzw. Applica-

tion Servers. Zusätzlich zum kompletten Pfad wird ein eventuell vorhandener Query-String weitergeleitet. Zudem führen die Parameter P und L dazu, dass der Request durch das Proxy-Modul geschickt (P) und die Prozessierung nach der jeweiligen Regel beendet (L) wird. Für den Betrieb weiterer Java-Anwendungen bzw. für einen AS ABAP können Sie die Einträge entsprechend erweitern.

Die Funktionalität des Rewriting-Moduls ist sehr umfangreich. Weitere Details finden Sie in der Apache-Dokumentation auf <http://httpd.apache.org/docs/2.2>.

Vervollständigung der grundlegenden Apache-Konfiguration und Verschlüsselung der Kommunikation

Prinzipiell sollte der initiale Zugriff so funktionieren, allerdings fehlen noch zwei Bausteine: Zum einen ist bisher nur der Kommunikationsweg zum Application Server berücksichtigt, der Kommunikationsweg der Antwort allerdings noch nicht. Zum anderen wird bisher der Host-Header noch nicht an den Application Server weitergeschickt.

1. Um diese fehlende Funktionalität zu erlangen, erweitern Sie die Konfiguration durch folgende Zeilen:

```
ProxyRequests Off
ProxyPreserveHost On
ProxyPassReverse /irj/
    http://<server>:<port>/irj/
ProxyPassReverse /logon/
    http://<server>:<port>/logon/
ProxyPassReverse /webdynpro/
    http://<server>:<port>/webdynpro/
```

Die Direktive `ProxyRequests Off` deaktiviert die Forward-Proxy-Funktionalität, da hier nur die Reverse-Proxy-Funktionalität benötigt wird. `ProxyPreserveHost` instruiert den Apache-Server, den Host-Header weiterzuleiten und ist essenziell für eine korrekte Funktion. Die letzten drei Konfigurationszeilen sind für den Kommunikationsweg vom Application Server zum Client verantwortlich.

2. Starten Sie den Apache-Server neu, und versuchen Sie, über `http://<ApacheHost>:<ApachePort>/irj/` auf ihn zuzugreifen. Sie sollten nun erkennen, dass Ihr Application Server über die URL des Apache-Servers aufgerufen werden kann.

3. Zusätzlich können Sie den Komfort für Ihre Anwender dadurch erhöhen, dass Sie Zugriffe auf die *irj*-Applikation ohne den abschließenden Schrägstrich (/) sowie den Zugriff auf den Apache-Webserver ohne Angabe der *irj*-Applikation durch Verwendung der *RedirectMatch*-Anweisung weiterleiten:

```
RedirectMatch 301 ^/$
    http://<ApacheHost>:<ApachePort>/
    irj/portal
RedirectMatch 302 ^/irj$
    http://<ApacheHost>:<ApachePort>/
    irj/portal
```

4. Um dieses Szenario etwas zu erweitern und somit praxisnäher zu gestalten, aktivieren Sie auf dem Apache-Server HTTPS und terminieren es dort. Terminierung bedeutet, dass ab diesem Punkt die weitere Kommunikation Richtung Application Server unverschlüsselt über HTTP stattfindet. Somit ist keine HTTPS-Konfiguration auf dem Application Server notwendig. Für den Betrieb von SSL mit Apache wird das Modul *mod_ssl* benötigt, das über folgende Konfiguration geladen wird, sofern es nicht mit Ihrem Server kompiliert wurde:

```
LoadModule ssl_module modules/mod_ssl.so
```

Weitere Details zur SSL-Konfiguration des Apache-Servers finden Sie in der Apache-Dokumentation. Für die Generierung des für die SSL-Kommunikation erforderlichen Serverzertifikats empfehlen wir Ihnen den Zertifikatsservice *SAP-SSL-Test-Server* (siehe Kasten »SAP-SSL-Test-Server-Zertifikat«).

In Listing 4.1 sehen Sie einen Auszug aus der Apache-Virtual-Host-Konfiguration für die SSL-Kommunikation. Auf die Parameter für die Apache-SSL-Konfiguration haben wir an dieser Stelle aus Gründen der Übersichtlichkeit verzichtet. Die Weiterleitungskonfiguration unterscheidet sich nur durch einen zusätzlichen Parameter (in Fettschrift formatiert) von der für HTTP notwendigen Konfiguration. Dieser Parameter (*RequestHeader set*) stellt unter Benutzung des Moduls *mod_headers* sicher, dass der Application Server über das Protokoll informiert wird, mit dem der Client mit dem Apache-Server kommuniziert.

SAP-SSL-Test-Server-Zertifikat

Über den SAP-SSL-Test-Server-Zertifikat-Service können Sie Testzertifikate für die SSL-Kommunikation generieren. Diese Zertifikate haben eine Gültigkeit von acht Wochen. Zunächst müssen Sie allerdings für Ihren Server einen privaten Serverschlüssel generieren.

1. Dies geschieht zum Beispiel mit *openssl* über das folgende Kommando auf Betriebssystemebene:
`openssl genrsa -out server.key -des3 1024`
2. Auf der Basis dieses Schlüssels kann eine Zertifikatsanfrage (*Certificate Signing Request, CSR*) über den folgenden *openssl*-Befehl generiert werden:
`openssl req -new -key server.key -out server.csr`
3. Bei der Abfrage der Zertifikatdetails müssen Sie sicherstellen, dass der Common Name (cn) dem voll qualifizierten Hostnamen (zum Beispiel *host.domain.com*) entspricht. Außerdem sollten Sie die Abfrage nach der E-Mail-Adresse nur mit **Enter** bestätigen, da eine gepflegte E-Mail-Adresse zu einem Fehler bei der Erstellung von SAP-Test-Zertifikaten führt.
4. Öffnen Sie nun die Seite <http://service.sap.com/tcs-ssl-test> bzw. <http://service.sap.com/ssltest>, und wählen Sie **Test it Now!**.
5. Öffnen Sie die Datei *server.csr* mit einem Texteditor, und kopieren Sie den Inhalt in das Feld **Enter data for public key**. Mit dem Button **Continue** starten Sie die Erstellung des Testzertifikats.
6. Das generierte Zertifikat erhalten Sie in einem Textfeld. Kopieren Sie den Inhalt in eine neue Textdatei, und speichern Sie diese unter dem Namen *server.crt*. Diese Datei stellt Ihr Serverzertifikat dar, das Sie für die SSL-Konfiguration des Apache-Servers benutzen können.

Die in Abschnitt 4.5, »Beispiel: Apache als Application Gateway für einen Application Server«, vorgestellte Konfiguration ist eine recht verbreitete Konfigurationsvariante. Abbildung 4.8 zeigt schematisch den Kommunikationsverlauf inklusive der übertragenen Header-Informationen (Host sowie ClientProtocol).

```

<VirtualHost _default_:443>
...
ServerName <server>:443
RewriteEngine On
RewriteLog logs/rewrite.log
RewriteLogLevel 3
RequestHeader set ClientProtocol HTTPS
RewriteRule ^/irj/(.*) http://<server>:<port>/irj/$1%{QUERY_STRING} [P,L]
RewriteRule ^/logon/(.*) http://<server>:<port>/logon/$1%{QUERY_STRING} [P,L]
RewriteRule ^/webdynpro/(.*) http://<server>:<port>/webdynpro/$1%{QUERY_STRING} [P,L]
ProxyRequests Off
ProxyPreserveHost On
ProxyPassReverse /irj/ http://<server>:<port>/irj/
ProxyPassReverse /logon/ http://<server>:<port>/logon/
ProxyPassReverse /webdynpro/ http://<server>:<port>/webdynpro/
RedirectMatch 301 ^/$ https://<ApacheHost>:<ApachePort>/irj/portal
RedirectMatch 302 ^/irj$ https://<ApacheHost>:<ApachePort>/irj/portal
...
</VirtualHost>

```

Listing 4.1 Apache-Konfiguration – Weiterleitung mit SSL-Terminierung

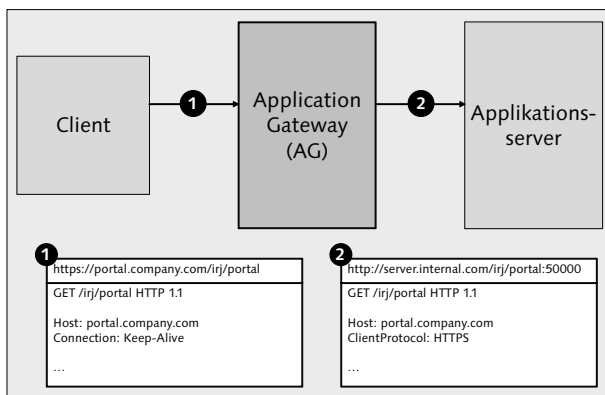


Abbildung 4.8 SSL-Terminierung am Application Gateway

4.6 Beispiel: Apache als Application Gateway für mehrere Application Server

Der in Abschnitt 4.3, »Konfigurationsanforderungen an Application Gateways«, angesprochene SAP-Hinweis 833960 beschreibt die Einschränkung, dass SAP keine Konfiguration unterstützt, bei der mehrere Application

Server unter einem externen Hostnamen zusammengefasst werden. Hintergrund ist, dass bei einer solchen Konfiguration, je nach Art der integrierten Anwendungen, URLs identisch sein können.

Das Beispiel in diesem Abschnitt zeigt daher, wie Sie einen Apache-Server konfigurieren, um mit einem einzigen Application Gateway mehrere Anwendungen zu schützen. Hierzu müssen Ihre Clients allerdings HTTP 1.1 verwenden, und Sie müssen auf dem Apache-Server eine Virtual-Host-Konfiguration erstellen. Dies bedeutet, dass mit einer einzigen IP-Adresse mehrere Hostnamen mit unterschiedlicher Konfiguration betrieben werden können.

In Listing 4.2 ist eine Konfiguration mit zwei Virtual Hosts dargestellt, die durch `NameVirtualHost` eingeleitet wird. Um die Virtual-Host-Konfigurationen zu unterscheiden, wird der Parameter `ServerName` (in Listing 4.2 fett markiert) pro Virtual Host unterschiedlich konfiguriert. Dadurch beeinflussen sich die beiden Anwendungen nicht gegenseitig. Die beiden Virtual Hosts definieren zudem ihre eigenen Rewrite-Regeln.

```

NameVirtualHost *:80
<VirtualHost *:80>
    ServerName <server1Extern>
    RewriteRule ^/irj/(.*) http://<server>:<port>/irj/$1?%{QUERY_STRING} [P,L]
    RewriteRule ^/logon/(.*) http://<server>:<port>/logon/$1?%{QUERY_STRING} [P,L]
    RewriteRule ^/webdynpro/(.*) http://<server>:<port>/webdynpro/$1?%{QUERY_STRING} [P,L]
    ProxyRequests Off
    ProxyPreserveHost On
    ProxyPassReverse /irj/ http://<server>:<port>/irj/
    ProxyPassReverse /logon/ http://<server>:<port>/logon/
    ProxyPassReverse /webdynpro/ http://<server>:<port>/webdynpro/
    RedirectMatch 301 ^/$ http://<server1Extern>:<ApachePort>/irj/portal
    RedirectMatch 302 ^/irj$ http://<server1Extern>:<ApachePort>/irj/portal
</VirtualHost>
<VirtualHost *:80>
    ServerName <server2Extern>
    RewriteRule ^/irj/(.*) http://<server2>:<port>/irj/$1?%{QUERY_STRING} [P,L]
    RewriteRule ^/logon/(.*) http://<server2>:<port>/logon/$1?%{QUERY_STRING} [P,L]
    RewriteRule ^/webdynpro/(.*) http://<server2>:<port>/webdynpro/$1?%{QUERY_STRING}
    ProxyRequests Off
    ProxyPreserveHost On
    ProxyPassReverse /irj/ http://<server2>:<port>/irj/
    ProxyPassReverse /logon/ http://<server2>:<port>/logon/
    ProxyPassReverse /webdynpro/ http://<server2>:<port>/webdynpro/
    RedirectMatch 301 ^/$ http://<server2Extern>:<ApachePort>/irj/portal
    RedirectMatch 302 ^/irj$ http://<server2Extern>:<ApachePort>/irj/portal
</VirtualHost>

```

Listing 4.2 Apache-Konfiguration – Virtual Hosts für mehrere Application Server

Index

A

- A-Gate 25
- ABAP-Workprozess 10
- absolute URL 47, 51
- Access Control List (ACL) 16, 60, 66
 - Administrationsberechtigung 61
 - Endbenutzerberechtigung 61, 63
 - KMC 66
 - Permission Viewer 69
 - Role-Assigner-Berechtigung 61
- Additional Application Server 75
- Administrationsberechtigung 61
- ADoW 13, 23
 - Client-Frontend 23, 125
 - GZip-Offloading 126
 - Komprimierung 125
 - Server-Frontend 23, 125
 - SSL-Offloading 126
 - Webcaching 125
- AJAX 110
- Akamai 125
- Akzeptanz 82, 104
- anonymer Benutzer 60, 69
- Antwortzeit 95
- Anwendungsperformance 102
- Anwendungssicherheit 16, 44
- Apache 16, 27, 47, 52
 - httpd.conf 52
 - LoadModule 52
 - mod_ext_filter 52
 - mod_headers 52, 54
 - mod_proxy 27, 52
 - mod_proxy_http 52
 - mod_rewrite 52
 - mod_ssl 52, 54
 - NameVirtualHost 56
 - ProxyPreserveHost 53
 - RedirectMatch 54
 - RequestHeader 54

- RewriteEngine 52
- RewriteLog 53
- RewriteRule 53
- Appliance 27, 47
- Application Delivery over WAN → ADoW
- Application Gateway 11, 21, 45
- Application-Integrator-iView 50
- Application Layer Firewall 45
- Application und Content Delivery 125
- Archivierung 83
- Ausfallzeit
 - geplante 77
 - ungeplante 72
- Authentifizierung 14, 57
 - Container-based Authentication 59
 - headerbasierte 58
 - UME-based Authentication 60

B

- Backup 77
- Backup und Restore 44
- Bandbreite 111
- Base64 57
- Basic Authentication 29
- Benchmark 85, 86, 93, 99
- Benutzerzahl 87
- Berechtigungseditor 60
- Betriebshandbuch 72
- Betriebskosten 18
- Betriebssystem
 - Virtualisierung 83
 - Sicherheit 43
- BEx Web 25
- BI-Bericht 25
- Blacklist 16
- Browser-Caching 15, 110
- Browser-Favoriten 29
- Business Server Pages 25

C

- Cache
 - HTTP-Cache 104
 - iView-Cache 104
 - KMC 104
 - Navigation-Cache 104
 - UME Cache 104
- CacheControl 112
- Caching 104, 110, 111
 - clientseitiges 112
 - serverseitiges 103
- Cascading Stylesheet 110
 - Sprites 110
- Certificate Signing Request → CSR
- CFE → ADoW
- Citrix 119
- Clickstream 95
- Client-Caching 112
- Client-Frontend → ADoW
- Client-Performance 101
- ClientProtocol 48
- Cluster-Gruppe 73, 74
- Cluster-Knoten 73, 74
- Cluster-Ressource 73
- Container-based Authentication 59
- Content Administration 60
- Content-Proxy 50
- Content Rewriting 51
- Cookie 48, 106
 - Spezifikation 48
- Corporate Directory 24
- Cross-Site Scripting → XSS
- Cryptographic Hardware Accelerator 52
- CSR 54
- Customizing 80

D

- Datenbank 102
 - Cluster 73
 - Connection-Pool 103

- Datenbanksicherheit 44
- Index 103
- Performance 102
- Reconnect 11
- Server 22
- Datendurchsatz 87
- demilitarisierte Zone → DMZ
- Denial of Service → DoS
- Dialoginstanz 75, 76
- Dialogschritt 9
- Directory-Server 44, 72
 - Directory-Server-Sicherheit 44
- DMZ 15, 19, 44
 - äußere 44
 - innere 44
- Domain Name Service (DNS) 18
 - Record 30
 - Resource Record Set 30
 - Server 29, 50
- DoS 11, 68, 97
- Dual Stack 73
- dynamischer Paketfilter 45

E

- E-Mail-Benachrichtigung 47
- E-Recruiting 5
- EFP 14, 105
- End-to-end-Analyse 115, 118
- Endbenutzerberechtigung 61, 63
- Enqueue Replication Server → ERS
- Enqueue-Server 73, 74
 - Locktable 73
- Enterprise SOA 5
- ERS 74, 76
- ESS-Szenario 90
- Exploit 16
- External Facing Portal → EFP

F

- Failover Target 76
- Favorit 29
- Fetch Mode 50
- Fileshare 75
- Firewall 19, 44
- First Line of Defense 11
- Form-based-Login 57
- Full Qualified Domain Name (FQDN) 47, 50

G

- Garbage Collection 97, 103
- GoingLive-Check 89

H

- Hacker 68
- Hardware
 - Ausfall 71
 - Cluster 73
 - Kosten 87
 - Load-Balancer 22
 - Planung 85
 - Sizing 77
 - Tausch 71
- Härtung 43
- Header-Login 58
- High-Availability-Setup 74
- Host-Header 48, 53, 106
- Hostname 50
 - virtueller 11
- HTML 9
- HTML-Rendering 24
- HTTP 9, 105, 109
 - Access Log 117
 - Analyse 115, 118
 - Basic Authentication 57
 - Cache 104
 - Client 105
 - Cookies 106
 - HTTPS 15, 57
 - HTTP Watch 116
 - Keep-Alive 106, 111
 - Persistent Connections 106
 - Pipelining 106
 - Provider Service 114, 117
 - Request 101, 105
 - Response 101, 105
 - Ressource 102
 - Server 105
- HTTP-Header 29, 58, 105
 - ClientProtocol 48, 54
 - Connection close 109
 - Host 48, 54, 106
 - Set-Cookie 48, 106
- HTTP-Provider-Service 34, 49, 112
- httpd.conf 52
- HttpServletRequest-Objekt 48
- Hypertext Markup Language → HTML
- Hypertext Transfer Protocol → HTTP

I

- ICA 119
- ICM 35, 112
- icm/server_port 39
- Independent Computing Architecture → ICA
- Infrastruktur
 - physikalische 86
 - technische 86, 93
- Infrastruktursicherheit 43
- Instanz 10
- Integrated Windows Authentication 58
 - Kerberos 58
 - NTLM 58
- Interception-Proxy 123
- Internet Connection Manager → ICM
- Internet Service Provider → ISP
- Internet Transaction Server → ITS
- IP-Spoofing 45
- IPsec 122
- ISO/OSI-Basisreferenzmodell 31
- ISP 14
- ITS 25
- iView-Caching 50, 104

J

- J2EE 59
- Java Authentication and Authorization Service (JAAS)
 - Login-Module 58
 - Login-Module-Flag 59
 - Login-Module-Option 59
 - Login-Module-Stack 59
- Java Connector → JCo
- Java-iView 24
- Java-Serverprozess 10
- Java Virtual Machine → JVM
- JavaScript
 - On-Demand-JavaScript 110
- JCo 24
- JVM 96, 103
 - Garbage Collection 103
 - Konfiguration 103

K

- Keep-Alive 106, 109
 - URL 33
- Kerberos 58
- Kernel 82
- Kioskszenario 121

Klonen 78
 KMC 18, 49, 66, 104
 ACL 66
 Alternative Host 50
 Berechtigung 67
 Notification 49
 Repository 66
 Repository Manager 113
 Ressource 66
 Service Permission 67
 Subscription 49
 URL Generator Service 49
 Knowledge Management
 Read-Only-Flag 113
 Repository 25
 Knowledge Management und
 Collaboration → KMC
 Komponententest 93
 Komprimierung 111, 114

L

LAN 108
 Landscaping 86
 Lastausgleich 72
 Lastgenerator 95
 Lasttest 86
 Lasttestprozess 94
 Lasttestsoftware 98
 Lastverteilung 27
 Latenz 108
 LDAP 22, 72
 LDAPS 24
 listPermissions 70
 Load Balancing 18, 22, 27, 46, 72
 clientbasiertes 28
 DNS-basiertes 29
 funktionales 35
 Microsoft Network Load Balancing 30
 serverbasiertes 31
 Loadtest 94
 Locktable 73
 Login-Modul 58
 Logon-Gruppe 35

M

Managed Services 125
 Mercury LoadRunner 96
 Message Server 37, 73, 74
 Microsoft Cluster Service → MSCS
 Microsoft Terminal Services 120

minimale Berechtigung 62
 Minimalprinzip 43
 MinumumGZipLength 114
 Mitarbeiter 67
 mod_ext_filter 52
 mod_headers 52, 54
 mod_proxy 27, 52
 mod_proxy_http 52
 mod_rewrite 52
 mod_ssl 52
 MSCS 72

N

NAT 31
 Navigation-Cache 104
 Network Address Translation → NAT
 Network Requirements 108
 Network-Sizing 85
 Netzwerk
 Bandbreite 111
 LAN 108
 Latenz 108
 Paketverlust 108
 Performance 105
 Qualität 14
 Round Trip Time 108
 Satellitenverbindung 108
 Sicherheit 43
 Umgebung 82
 WAN 108
 Netzwerkzone 6, 19
 NeverCompressed 114
 niping 115
 NTLM 58

O

offener Port 43
 öffentliches Portal 5
 Offloading 47, 52
 GZip 126
 SSL 126
 On-Demand-JavaScript 110
 Online-Bewerbermanagement 5
 Online-Shop 5
 Open Source 47
 Openssl 54

P

Paketfilter 44, 46
 dynamischer 45

statischer 44
 Paketverlust 108
 PAM 35
 Patchen 71
 PCD 60, 82
 Peak-Load-Sizing 88
 Peak-Load-Szenario 82, 93
 perfmon 96
 Performance 86, 101
 Anwendung 101, 102
 Client 101
 Datenbank 102
 Hardwareausstattung 101
 Netzwerk 101, 105
 Server 101, 102
 Webbrowser-Konfiguration 101
 Performance-Messung
 Client 116
 Netzwerk 115
 Server 117
 Permission Viewer 69
 Personalisierung 61, 62
 physische Sicherheit 44
 ping 115
 Pipelining 106
 PKI 57
 Planned Downtime 71, 77
 Policy Configuration 58
 Polling 76
 Portal-Content 80
 Portal Content Catalog 61
 Portal Content Directory → PCD
 Portal-Service
 Content Fetching 113
 PortalAnywhere.Go 68
 Positivliste 16
 Primary Application Server 75, 76
 Product Availability Matrix → PAM
 Profildatei 39
 Profilparameter
 ProtocolHeaderName 49
 Protokoll-Switch 48
 Proxy 46
 Proxy-Server 19
 Prozessorlast 98
 Public Key Infrastructure → PKI

Q

Quellsystem 81
 Quick Sizer 86, 87, 90

R

Read-only-Modus 82
 Rechenzentrum 44
 Redirect 47, 50
 Referenzinstallation 88
 Regressionstest 89, 99
 reguläre Ausdrücke 47
 relative URL 47
 Release-Upgrade 77
 Releasewechsel 99
 Remote Desktop Connection 120
 Remote Desktop Protocol 120
 Remote Function Call → RFC
 Remote-Standort 20
 Rendering 24
 Request 105
 Resource Record Set 30
 Response 105
 Ressourcen-Management 82
 Restore 80, 83
 Reverse-Proxy 19, 46
 Zugriff 11
 Rewrite-Regel 51
 RFC 22
 Role-Assigner-Berechtigung 61
 Rollback 79, 83
 Round Robin 34, 35
 Weighted 34, 35
 Round Trip Time → RTT
 Router 44
 rstatd 96
 RTT 30, 108, 114

S

Safety Level 64
 Sales and Distribution Benchmark → SD
 SAP Active Global Support 89, 94
 SAP Application Delivery over WAN 23, 125
 SAP Application Performance Standard → SAPS
 SAP Central Services → SCS
 SAP Config Tool 66
 sap-contextid 35
 SAP GoingLive-Check 99
 SAP GUI
 für HTML 25
 für Java 25
 für Windows 25
 SAP Internet Connection Manager 112

SAP-Kernel 36
 SAP Management Console (SAP MC) 40
 SAP Microsoft Management Console (SAP MMC) 40
 SAP NetWeaver Application Server Java
 Client-Caching 112
 HTTP-Provider-Service 112, 114, 117
 Keep-Alive 111
 Komprimierung 114
 SAP Solution Manager 118
 SAP Solution Manager Diagnostics 115, 118
 SAP-SSL-Test-Server-Zertifikat-Service
 SAP Volume Test Optimization Onsite 94
 SAP Web Dispatcher 16, 35, 46, 72, 112
 SapCacheControl 112
 SAPINST 72
 saplb-Cookie 34
 sapmnt 73
 SAPS 85
 sapstartsrv 40
 Satellitenverbindung 108
 Scale-in 10
 Scale-out 10
 Schattensystem 78, 80
 Schlüssellänge 43
 Schutzziel 43, 47
 Script-Kiddies 68
 SCS 73
 SD 85
 SDM 73
 Second Level of Security 66
 Secure Network Communications → SNC
 Secure Sockets Layer → SSL
 Secure Store 43
 SecureId 122
 Security Audit 68
 Security-Audit 16
 Security Audit Logging 44
 Security by obscurity 16
 Security Optimization Service 68
 Security-Provider-Service 58
 Security Zone 63
 -Dcom.sap.nw.sz 66
 Berechtigungsvergabe 65
 Prüfung 68
 SafetyLevel 64
 Second Level of Security 66
 Security Zone Enforcement 65
 SecurityArea 64
 Vendor 64

Self-Service-Funktion 5
 Server-Frontend → ADoW
 Serverperformance 102, 117
 Serverprozess 10
 serverseitiges Caching 103
 Service Level Agreement → SLA
 Session-Information 9
 Session-Persistenz 10
 Session Stickiness 10, 27, 32
 SFE → ADoW
 Sicherheit 6, 10
 Sicherheitslücke 16
 Sicherheitsuntersuchung 68
 simple_round_robin 38
 Single Points of Failure → SPOF
 Single Sign-On 71
 Sizing 85
 benutzerbasiertes 88
 durchsatzbasiertes 88
 Projekt 18
 Prozess 85, 87
 T-Shirt-Größe 89
 Skalierbarkeit 9, 85
 Skripterstellung 95
 Skriptrekorder 96
 SLA 71, 78
 Smartcard 58
 SNC 16
 Sniffer
 Browser-Plug-in 109
 Ethereal 108
 Wireshark 108
 Social Engineer 68
 Software-Cluster 73
 Software-Load-Balancer 22
 Sperrtabelle 73, 75
 Split Mirror 77
 SPNego 58
 SPOF 72, 75
 SQL Injection 68
 Squid 126
 SSL 16, 57, 117, 122
 SSL-Terminierung 32, 35, 48
 Startseite 105
 Stateful Inspection 45
 Stateful Packet Filter 45
 Stresstest 94
 Such- und Klassifizierungsmaschine 23
 Support Package Stack 77
 Switch 83
 Switchover-Gruppe 76

Switchover-Lösung 72, 73
 System Copy Tool 78
 Systemadministration 60
 Systemkopie 78, 80
 Systemobjekt 62
 Systemrollentausch 83

T

Transmission Control Protocol (TCP) 31,
 46, 106, 109
 Slowstart 106
 Three Way Handshake 109
 Terminal-Server 14, 21, 119
 Einsatzszenario 121
 Hardwareanforderung 120
 Nachteil 120
 Performance 120
 Sicherheit 120
 Vorteil 120
 Terminal Services Client 120
 Thinktime 87, 96
 Three Way Handshake 109
 Transaktion 9
 Transaktionalität 9
 Transport 77
 Transportverschlüsselung 57
 TREX → Such- und Klassifizierungs-
 maschine

U

UME 57
 UME Cache 104
 UME-based Authentication 60
 Unified Rendering Framework → UR
 Uniform Resource Locator → URL
 Unplanned Downtime 71
 UR 10
 URL 16
 absolute 47
 relative 47
 URL-Filterregel 41
 URL Generator Service 49

URL-Integration 50
 URL-iView 50, 63
 clientseitiger 50
 serverseitiger 50
 User Management Engine → UME

V

Vererbung
 von Berechtigungen 62
 Verfügbarkeit 71
 Verschlüsselung 43
 Verzeichnisdienst 22, 72
 Virtual Host 54, 55
 Virtual Private Network → VPN
 Virtualisierung 81
 Visual Administrator 49, 58, 111, 112,
 117
 Volumentest 94
 Voraussetzung 7
 VPN 14, 20, 121
 Client 122
 End-to-End 121
 End-to-Site 122
 Gateway 122
 Sicherheit 122
 Site-to-Site 121
 VPN-Gateway 122
 VPN-Tunnel 122

W

W-Gate 25
 WAM 58
 WAN 108
 WAN-Beschleunigung 123
 Application und Content Delivery 125
 Hardware 124
 SAP Application Delivery over WAN
 125
 Web Proxy Cache 123
 Wartungsarbeit 72
 Watchdog 41, 72
 Web 2.0 5, 110

Web Access Management → WAM 58
 Web Cache 23, 41, 123
 Web Dispatcher 72
 Web Dynpro 10, 27, 53, 59, 60
 Web-Infrastruktur 6
 Web Proxy Cache 123
 Kosten 124
 Nachteil 124
 Performance 124
 Sicherheit 124
 Vorteil 124
 Webbrowser
 Cache-Konfiguration 110
 Firefox 106
 Internet Explorer 106
 WebGUI 25
 Webproxy 46
 Webservice 24
 Webzugriffsszenario 13
 weighted_round_robin 38
 Whitelist 16
 Workload 85
 Workprozess 10
 Konzept 11
 wpdispmon 38

X

X Window System 119
 X.509-Client-Zertifikat 41, 48, 57
 X11 119
 XSS 68

Z

ZBV 57
 ZDM 80
 Zeiterfassung 11
 zentrale Benutzerverwaltung → ZBV
 Zentralinstanz 75, 76
 Zero Downtime Maintenance → ZDM
 Zertifikat 43, 47, 54
 zusätzlicher Applikationsserver 76