



Sun Java™ System

# Sun Java Enterprise System 2005Q1 Technischer Überblick

---

Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
USA

Teilenr.: 819-1923

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, USA. Alle Rechte vorbehalten.

Sun Microsystems, Inc., hat Rechte in Bezug auf geistiges Eigentum an der Technologie, die in dem in diesem Dokument beschriebenen Produkt enthalten ist. Im Besonderen und ohne Einschränkung umfassen diese Ansprüche in Bezug auf geistiges Eigentum eines oder mehrere der unter <http://www.sun.com/patents> aufgelisteten Patente und eines oder mehrere Patente oder Anwendungen mit laufendem Patent in den USA und in anderen Ländern.

DIESES PRODUKT ENTHÄLT VERTRAULICHE INFORMATIONEN UND GESCHÄFTS-GEHEIMNISSE VON SUN MICROSYSTEMS, INC. VERWENDUNG, OFFENLEGUNG ODER REPRODUKTION IST OHNE VORHERIGE SCHRIFTLICHE GENEHMIGUNG DURCH SUN MICROSYSTEMS, INC., NICHT GESTATTET.

Rechte der US-Regierung - Kommerzielle Software. Regierungsbutzer unterliegen der standardmäßigen Lizenzvereinbarung von Sun Microsystems, Inc., sowie den anwendbaren Bestimmungen der FAR und ihrer Zusätze.

Diese Ausgabe kann von Drittanbietern entwickelte Bestandteile enthalten.

Teile dieses Produkts können von Berkeley BSD Systems abgeleitet sein, lizenziert durch die University of California. UNIX ist ein eingetragenes Warenzeichen in den USA und anderen Ländern und exklusiv durch X/Open Company, Ltd. lizenziert.

Sun, Sun Microsystems, das Sun-Logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, das Duke-Logo, das Java-Kaffeetassenlogo, das Solaris-Logo, das SunTone Certified-Logo und das Sun ONE-Logo sind Warenzeichen oder eingetragene Warenzeichen von Sun Microsystems, Inc., in den USA und anderen Ländern.

Alle SPARC-Warenzeichen werden unter Lizenz verwendet und sind Warenzeichen oder eingetragene Warenzeichen von SPARC International, Inc., in den USA und anderen Ländern. Produkte, die SPARC-Warenzeichen aufweisen, basieren auf der von Sun Microsystems, Inc., entwickelten Architektur.

Legato und das Legato-Logo sind eingetragene Warenzeichen und Legato NetWorker ist ein Warenzeichen bzw. ein eingetragenes Warenzeichen von Legato Systems, Inc. Das Netscape Communications Corp-Logo ist ein Warenzeichen bzw. ein eingetragenes Warenzeichen der Netscape Communications Corporation.

Die grafische Benutzeroberfläche von OPEN LOOK und Sun™ wurden von Sun Microsystems, Inc., entwickelt, für die entsprechenden Benutzer und Lizenznehmer. Sun erkennt die Pionierleistung von Xerox bei der Ausarbeitung und Entwicklung des Konzepts von visuellen oder grafischen Benutzeroberflächen für die Computerindustrie an. Sun ist Inhaber einer einfachen Lizenz von Xerox für die Xerox Graphical User Interface (grafische Benutzeroberfläche von Xerox). Mit dieser Lizenz werden auch die Sun-Lizenznehmer abgedeckt, die grafische OPEN LOOK-Benutzeroberflächen implementieren und sich ansonsten an die schriftlichen Sun-Lizenzvereinbarungen halten.

Produkte, die in diesem Diensthandbuch beschrieben sind, und die in diesem Handbuch enthaltenen Informationen unterliegen den Gesetzen der US-Exportkontrolle und können den Export- oder Importgesetzen anderer Länder unterliegen. Die Verwendung im Zusammenhang mit Nuklear-, Raketens-, chemischen und biologischen Waffen, im nuklear-maritimen Bereich oder durch in diesem Bereich tätige Endbenutzer, direkt oder indirekt, ist strengstens untersagt. Der Export oder Rückexport in Länder, die einem US-Embargo unterliegen, oder an Personen und Körperschaften, die auf der US-Exportausschlussliste stehen, einschließlich (jedoch nicht beschränkt auf) der Liste nicht zulässiger Personen und speziell ausgewiesener Staatsangehöriger, ist strengstens untersagt.

DIE DOKUMENTATION WIRD IN DER VERFÜGBAREN FORM ZUR VERFÜGUNG GESTELLT UND ALLE AUSDRÜCKLICHEN ODER STILLSCHWEIGENDEN BEDINGUNGEN, ANGABEN UND GARANTIEEN, INKLUSIVE ALLER STILLSCHWEIGENDEN GARANTIEEN BEZÜGLICH HANDELSÜBLICHKEIT, EIGNUNG ZU EINEM BESTIMMTEN ZWECK ODER MÄNGELGEWÄHR, SIND VON DER HAFTUNG AUSGESCHLOSSEN, AUSSER EIN SOLCHER AUSSCHLUSS WIRD ALS RECHTSWIDRIG BEFUNDEN.

# Inhalt

<b>Liste der Tabellen</b> .....	<b>7</b>
<b>Liste der Abbildungen</b> .....	<b>9</b>
<b>Vorwort</b> .....	<b>11</b>
Zielgruppe .....	12
Verwenden der Dokumentation .....	12
Konventionen .....	14
Ressourcen im Internet .....	14
So können Sie Probleme mitteilen .....	15
Kommentare sind willkommen .....	15
<b>Kapitel 1 Einführung in Java Enterprise System</b> .....	<b>17</b>
Wozu benötigen Sie Java Enterprise System? .....	18
Java Enterprise System-Komponenten .....	21
Systemdienstkomponenten .....	22
Dienstqualitätskomponenten .....	23
Verfügbarkeitskomponenten .....	24
Zugriffskomponenten .....	24
Verwaltungskomponenten .....	25
Gemeinsam genutzte Komponenten .....	26
Arbeiten mit Java Enterprise System .....	27
Java Enterprise System Lösungslebenszyklus .....	27
Java Enterprise System Einführungsszenarien .....	30
In diesem Kapitel enthaltene Schlüsselbegriffe .....	32

<b>Kapitel 2 Lösungsarchitekturen von Java Enterprise System</b> .....	<b>35</b>
Java Enterprise System-Architekturframework .....	36
Dimension 1: Infrastrukturdienstabhängigkeiten .....	37
Infrastrukturdienstebenen .....	37
Java Enterprise System-Infrastrukturdienstkomponenten .....	41
Java Enterprise System Infrastrukturdienstabhängigkeiten .....	42
Dimension 2: Logische Schichten .....	43
Beschreibung der logischen Schichten .....	44
Logische und physische Unabhängigkeit .....	45
Auf Systemkomponenten angewendete geschichtete Architektur .....	46
Dimension 3: Dienstqualität .....	47
Dienstqualitäten .....	48
Java Enterprise System-Dienstqualitätskomponenten .....	49
Sun Cluster-Software .....	50
Synthese der drei Architekturdimensionen .....	52
Beispiel für Java Enterprise System-Lösungsarchitektur .....	53
Szenario der Unternehmenskommunikation .....	53
Logische Architektur des Beispielszenarios .....	54
Bereitstellungsarchitektur des Beispielszenarios .....	56
In diesem Kapitel enthaltene Schlüsselbegriffe .....	57
<b>Kapitel 3 Integrationsfunktionen von Java Enterprise System</b> .....	<b>59</b>
Das integrierte Installationsprogramm von Java Enterprise System .....	60
Überprüfung bereits vorhandener Software .....	61
Überprüfung der Abhängigkeiten .....	61
Erstkonfiguration .....	62
Deinstallation .....	62
Integrierte Identitäts- und Sicherheitsdienste .....	62
Einzelidentität .....	63
Authentifizierung und Single Sign-On .....	64
Authentifizierung .....	65
Single Sign-On .....	66
Autorisierung .....	66
In diesem Kapitel enthaltene Schlüsselbegriffe .....	68

<b>Kapitel 4 Aufgaben des Lösungslebenszyklus von Java Enterprise System</b> .....	<b>69</b>
Bereitstellungsvorbereitung .....	71
Bereitstellung .....	72
Bereitstellungskonzept .....	73
Bereitstellungsarchitektur .....	73
Implementierungsspezifikationen .....	74
Implementierungspläne .....	75
Bereitstellungsimplementierung .....	75
Einrichten der Hardware .....	75
Installation, Aktualisierung und Migration der Software .....	76
Konfiguration und Anpassung des Systems .....	76
Entwicklung und Integration .....	77
Testen von Prototypen und Piloten .....	77
Produktionseinsatz .....	78
Bereitstellungsnachbereitung .....	78
In diesem Kapitel enthaltene Schlüsselbegriffe .....	79
<b>Anhang A Referenzliste: Komponenten von Java Enterprise System</b> .....	<b>81</b>
Beschreibungen der Systemdienstkomponenten .....	82
Sun Java System Access Manager 6 2005Q1 .....	82
Sun Java System Application Server Enterprise Edition 8 2005Q1 .....	83
Sun Java System Calendar Server 6 2005Q1 .....	84
Sun Java System Directory Server 5 2005Q1 .....	84
Sun Java System Instant Messaging 7 2005Q1 .....	84
Sun Java System Message Queue 3 2005Q1 .....	85
Sun Java System Messaging Server 6 2005Q1 .....	86
Sun Java System Portal Server 6 2005Q1 .....	86
Sun Java System Web Server 6.1 2005Q1 .....	87
Beschreibung der Dienstqualitätskomponenten .....	87
Beschreibung der Verfügbarkeitskomponenten .....	87
Sun Cluster 3.1 9/04 und Sun Cluster Agents .....	88
High Availability Session Store 2005Q1 .....	88
Beschreibung der Zugriffskomponenten .....	89
Sun Java System Communications Express 2005Q1 .....	89
Sun Java System Connector für Microsoft Outlook 6 2005Q1 .....	89
Sun Java System Directory Proxy Server 5 2005Q1 .....	90
Sun Java System Portal Server Secure Remote Access 6 2005Q1 .....	90
Beschreibung der Verwaltungskomponenten .....	91
Sun Java System Administration Server (und Console) 5 2005Q1 .....	91
Sun Java System Directory Preparation Tool 2005Q1 .....	91
Sun Java System Delegated Administrator 6 2005Q1 .....	91
Sun Remote Services Net Connect 2005Q1 .....	92
Gemeinsam genutzte Komponenten .....	92
<b>Index</b> .....	<b>95</b>



# Liste der Tabellen

Tabelle 1	Java Enterprise System Dokumentation .....	13
Tabelle 2	Schriftartkonventionen .....	14
Tabelle 1-1	Java ES-Systemdienstkomponenten .....	22
Tabelle 1-2	Java ES-Verfügbarkeitskomponenten .....	24
Tabelle 1-3	Java ES-Zugriffskomponenten .....	24
Tabelle 1-4	Java ES-Verwaltungskomponenten .....	25
Tabelle 1-5	Java ES-Benutzerkategorien für Aufgaben des Lebenszyklus .....	29
Tabelle 1-6	Aspekte unterschiedlicher Java ES-Einführungsszeanrien .....	31
Tabelle 2-1	Beziehungen zwischen Java ES -Systemdienstkomponenten .....	42
Tabelle 2-2	Dienstqualitäten mit Auswirkung auf die Lösungsarchitektur .....	48
Tabelle 2-3	Dienstqualitätskomponenten und die beeinflussten Systemqualitäten .....	49
Tabelle 2-4	Zusammenfassung der Geschäftsanforderungen: Kommunikationsszenario .....	54



# Liste der Abbildungen

Abbildung 1-1	Erforderliche Unterstützung für verteilte Unternehmensanwendungen	19
Abbildung 1-2	Kategorien von Java Enterprise System-Komponenten	21
Abbildung 1-3	Phasen des Lebenszyklus einer Lösung und Benutzerkategorien	28
Abbildung 2-1	Dimensionen der Java ES-Lösungsarchitektur	37
Abbildung 2-2	Dimension 1: Infrastrukturdienstebenen	38
Abbildung 2-3	Java ES Systemdienstkomponenten	41
Abbildung 2-4	Dimension 2: Logische Schichten für verteilte Unternehmensanwendungen	43
Abbildung 2-5	Messaging Server: Beispiel für eine Schichtenarchitektur	47
Abbildung 2-6	Verfügbarkeitskonzept mit Sun Cluster-Knoten	51
Abbildung 2-7	Logische Architektur des Unternehmenskommunikationsszenarios	55
Abbildung 3-1	Ein einzelner Benutzereintrag unterstützt viele Dienste	64
Abbildung 3-2	Authentifizierungssequenz	65
Abbildung 3-3	Autorisierungssequenz	67
Abbildung 4-1	Aufgaben des Lösungslebenszyklus	70
Abbildung 4-2	Spezifizieren eines Bereitstellungsszenarios	72
Abbildung 4-3	Umsetzen eines Bereitstellungsszenarios in eine Bereitstellungsarchitektur	73



# Vorwort

*Das Handbuch Sun Java™ Enterprise System Technischer Überblick* stellt die technischen und konzeptionellen Grundlagen von Java Enterprise System vor. Darüber hinaus werden die Komponenten, die Architektur, die Vorgänge und die Funktionen von Java Enterprise System beschrieben.

Dieser Überblick versucht, die im Java Enterprise System-Dokumentationssatz verwendeten technischen Konzepte und die zugehörige Terminologie zu erklären. Wichtige technische Begriffe werden in jedem Kapitel im Abschnitt „Schlüsselbegriffe“ erläutert. Dort wird beschrieben, wie diese Begriffe im Java Enterprise System-Kontext verwendet werden.

Dieses Vorwort enthält die folgenden Abschnitte:

- „Zielgruppe“ auf Seite 12
- „Verwenden der Dokumentation“ auf Seite 12
- „Konventionen“ auf Seite 14
- „Ressourcen im Internet“ auf Seite 14
- „So können Sie Probleme mitteilen“ auf Seite 15
- „Kommentare sind willkommen“ auf Seite 15

## Zielgruppe

Das Handbuch *Java Enterprise System Technical Overview* ist für Benutzer gedacht, die Softwarelösungen auf Basis von Java Enterprise System entwerfen, bereitstellen oder verwalten. Zu dieser großen Zielgruppe gehören beispielsweise Geschäftsanalysten, Systemarchitekturdesigner, Außendiensttechniker und Systemadministratoren.

Zum Verständnis des Handbuchs *Java Enterprise System Technical Overview* sind Kenntnisse der folgenden Technologien erforderlich:

- Allgemeine Netzwerkkonzepte
- Sicherheitsgrundlagen für Authentifizierung und Autorisierung
- Java-Sprache, Java 2 Standard Edition-Komponenten und Java 2 Enterprise Edition-Komponenten

## Verwenden der Dokumentation

Die Java Enterprise System-Handbücher sind als Online-Dateien im PDF-Format (Portable Document Format) und HTML-Format (Hypertext Markup Language) verfügbar. Beide Formate können mit unterstützenden Technologien von Benutzern mit Behinderungen gelesen werden. Unter folgender Adresse können Sie auf die Sun<sup>TM</sup>-Dokumentations-Website zugreifen:

<http://docs.sun.com>.

Zugriff auf die Java Enterprise System-Dokumentation:

<http://docs.sun.com/prod/entsys.05q1>

In der nachfolgenden Tabelle sind die Handbücher auf Systemebene aufgeführt, die Bestandteil der Java Enterprise System-Dokumentationssammlung sind. In der linken Spalte werden der Name sowie der Speicherort der Teilenummer der einzelnen Dokumente angegeben. In der rechten Spalte wird der allgemeine Inhalt des Dokuments beschrieben.

**Tabelle 1** Java Enterprise System Dokumentation

Dokument	Inhalt
<i>Java Enterprise System Release Notes</i> <a href="http://docs.sun.com/doc/819-0057">http://docs.sun.com/doc/819-0057</a>	Enthält die neuesten Informationen über Java Enterprise System, einschließlich bekannter Probleme. Zudem verfügen die Komponenten über ihre eigenen Versionshinweise.
<i>Java Enterprise System Documentation Roadmap</i> <a href="http://docs.sun.com/doc/819-0055">http://docs.sun.com/doc/819-0055</a>	Enthält Beschreibungen der Dokumentation zu Java Enterprise System. Enthält Verknüpfungen zur Dokumentation der Komponenten.
<i>Java Enterprise System Technical Overview</i> <a href="http://docs.sun.com/doc/819-0061">http://docs.sun.com/doc/819-0061</a>	Stellt die technischen und konzeptionellen Grundlagen von Java Enterprise System vor. Beschreibt Komponenten, Architektur, Vorgänge und Funktionen.
<i>Java Enterprise System Deployment Planning Guide</i> <a href="http://docs.sun.com/doc/819-0058">http://docs.sun.com/doc/819-0058</a>	Bietet eine Einführung in die Planung und Konzeption von Bereitstellungslösungen für Unternehmen auf der Grundlage von Java Enterprise System. Stellt grundlegende Konzepte und Prinzipien der Bereitstellungsplanung und -konzeption vor, behandelt den Lösungslebenszyklus und bietet anspruchsvolle Beispiele und Strategien für die Planung von Lösungen auf der Grundlage von Java Enterprise System.
<i>Java Enterprise System User Management Guide</i> <a href="http://docs.sun.com/doc/817-5761">http://docs.sun.com/doc/817-5761</a>	Unterstützt Sie bei der Planung, Bereitstellung und Verwaltung von Informationen der Benutzer Ihrer Java Enterprise System-Lösung. Ergänzt das Handbuch <i>Java Enterprise System Deployment Planning Guide</i> durch die Beschreibung der in den einzelnen Phasen des Lösungslebenszyklus bestehenden Problemen der Benutzerverwaltung.
<i>Java Enterprise System Deployment Example Series: Evaluation Scenario</i> <a href="http://docs.sun.com/doc/819-0059">http://docs.sun.com/doc/819-0059</a>	Beschreibt die Installation von Java Enterprise System auf einem System sowie das Herstellen einer Reihe wichtiger, freigegebener Netzwerkdienste und das Einrichten der Benutzerkonten, auf die Sie über die von Ihnen eingerichteten Dienste zugreifen können.
<i>Java Enterprise System-Installationshandbuch</i> <a href="http://docs.sun.com/doc/819-0056">http://docs.sun.com/doc/819-0056</a>	Führt Sie durch die Installation von Java Enterprise System für die Betriebssysteme Solaris™ und Linux. Zeigt, wie Sie die Komponenten auswählen, die Sie installieren möchten, wie Sie die von Ihnen installierten Komponenten konfigurieren und wie Sie überprüfen, ob die konfigurierten Komponenten ordnungsgemäß funktionieren.
<i>Java Enterprise System Upgrade and Migration Guide</i> <a href="http://docs.sun.com/doc/819-0062">http://docs.sun.com/doc/819-0062</a>	Bietet die für die Aktualisierung von Java Enterprise System für die Betriebssystemumgebungen von Solaris™ und Linux notwendigen Informationen und Anleitungen.
<i>Java Enterprise System Glossary</i> <a href="http://docs.sun.com/doc/816-6873">http://docs.sun.com/doc/816-6873</a>	Definiert die Begriffe, die in der Java Enterprise System-Dokumentation verwendet werden.

Neben den in dieser Tabelle aufgeführten systembezogenen Dokumenten umfasst der Dokumentationssatz von Java Enterprise System auch spezifische Dokumentationen für die einzelnen Komponenten von Java Enterprise System. Einzelheiten finden Sie im Handbuch *Java Enterprise System Documentation Roadmap*.

## Konventionen

In der folgenden Tabelle werden die in diesem Handbuch verwendeten Schriftartkonventionen beschrieben.

**Tabelle 2** Schriftartkonventionen

Schriftart	Bedeutung	Beispiele
AaBbCc123 (nichtproportional)	API und Sprachelemente, HTML-Tags, Website-URLs, Befehlsnamen, Dateinamen, Verzeichnispfadnamen, Bildschirmcomputerausgaben, Beispielcode.	Bearbeiten Sie Ihre <code>.login</code> -Datei. Verwenden Sie <code>ls -a</code> zum Auflisten aller Dateien. % Sie haben eine E-Mail erhalten.
<i>AaBbCc123</i> (kursiv)	Buchtitel Neue Wörter oder Begriffe Hervorzuhebende Wörter Befehlszeilenvariablen, die durch tatsächliche Namen oder Werte ersetzt werden müssen	Lesen Sie im <i>Benutzerhandbuch</i> Kapitel 6. Diese Optionen werden <i>class</i> -Optionen genannt. Sie <i>müssen</i> als „superuser“ angemeldet sein, um diesen Vorgang durchzuführen. Die Datei befindet sich im Verzeichnis <code>install-dir/bin</code> .

## Ressourcen im Internet

Unter folgender Adresse erhalten Sie Informationen über Java Enterprise System und seine Produktkomponenten:

<http://www.sun.com/software/javaenterprisesystem/index.html>

## So können Sie Probleme mitteilen

Wenn Sie mit Java Enterprise System Probleme haben, wenden Sie sich an die Kundenunterstützung von Sun. Dazu stehen Ihnen folgende Möglichkeiten zur Verfügung:

- Sun-Softwaresupport unter:

<http://www.sun.com/service/sunone/software>

Auf dieser Website finden Sie Links zur Knowledge Base, zum Online Support Center, zum ProductTracker wie auch zu Wartungsprogrammen und Kontaktinformationen für die Kundenunterstützung.

- Die auf Ihrem Wartungsvertrag angegebene Telefonnummer.

Damit wir Sie optimal beraten können, halten Sie bitte die folgenden Informationen bereit, wenn Sie sich an die Kundenunterstützung wenden:

- Beschreibung des Problems, einschließlich der Situation, in der das Problem auftrat, sowie seine Auswirkungen auf Ihre Arbeit.
- Rechnertyp, Betriebssystem- und Produktversion, einschließlich sämtlicher Patches und anderer Software, die mit dem Problem in Zusammenhang stehen könnten.
- Zur Nachvollziehung des Problems eine ausführliche Beschreibung der einzelnen Schritte und Vorgehensweisen, die zu dem Problem geführt haben.
- Sämtliche Fehlerprotokolle oder Kernspeicherauszüge.

## Kommentare sind willkommen

Sun möchte seine Dokumentation laufend verbessern. Ihre Kommentare und Vorschläge sind daher immer willkommen.

Zum Mitteilen Ihrer Kommentare rufen Sie <http://docs.sun.com> auf und klicken Sie auf „Send Comments“. Geben Sie im Online-Formular den Dokumenttitel und die Teilenummer an. Die Teilenummer ist eine 7-stellige oder 9-stellige Zahl, die Sie auf der Titelseite des Handbuchs oder am Anfang des Dokuments finden. Der Titel dieses Buchs lautet beispielsweise *Sun Java Enterprise System Technical Overview* und die Teilenummer ist 819-1923.

Kommentare sind willkommen

# Einführung in Java Enterprise System

Sun Java™ Enterprise System (Java ES) besteht aus einem Satz von Softwarekomponenten, die Dienste bieten, die bei der Unterstützung verteilter Anwendungen in Unternehmensstärke in einer Netzwerk- oder Internetumgebung benötigt werden. Solche Anwendungen werden in diesem Buch als verteilte Unternehmensanwendungen bezeichnet.

Java Enterprise System ist gleichzeitig eine Sun-Softwareversion und eine Zustellungsmethode sowie eine Geschäfts- und Preisstrategie. Der Schwerpunkt dieses Buchs liegt jedoch auf den Softwarekomponenten von Java Enterprise System und den dadurch bereitgestellten Diensten.

In diesem Kapitel werden das Java Enterprise System und die Aufgaben vorgestellt, die bei Verwendung des Systems ausgeführt werden. Es deckt folgende Themen ab:

- „Wozu benötigen Sie Java Enterprise System?“ auf Seite 18
- „Java Enterprise System-Komponenten“ auf Seite 21
- „Arbeiten mit Java Enterprise System“ auf Seite 27
- „In diesem Kapitel enthaltene Schlüsselbegriffe“ auf Seite 32

# Wozu benötigen Sie Java Enterprise System?

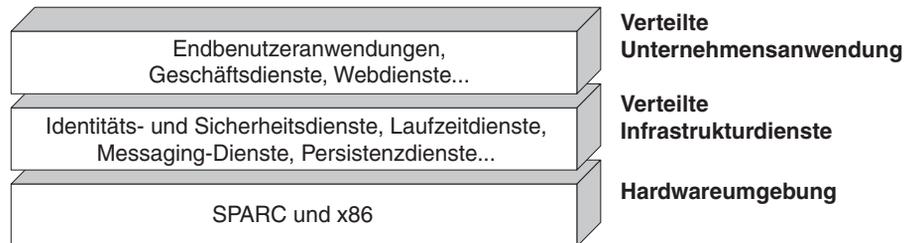
Die Geschäftsanforderungen von heute fordern Softwarelösungen, die über eine Netzwerk- oder Internetumgebung hinweg verteilt sind und sehr viel Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit bieten.

Java Enterprise System bietet Infrastrukturdienste, die benötigt werden, um *Verteilte Unternehmensanwendungen* zu unterstützen, die in der Regel folgende Merkmale aufweisen:

- **Verteilt:** Die Anwendung umfasst interagierende Software-*Komponenten*, die über eine vernetzte Umgebung hinweg bereitgestellt werden, zu der geografisch andere Standorte gehören können. Diese verteilten Komponenten, die auf den verschiedenen Computern der Umgebung ausgeführt werden, arbeiten zusammen, um spezielle Geschäftsfunktionen für die *Endbenutzer* und andere Geschäftsanwendungen bereitzustellen.
- **Unternehmensstärke:** Umfang und Größe der Anwendung erfüllen die Anforderungen einer Produktionsumgebung oder eines Internetdiensteanbieters. Die Anwendung erstreckt sich in der Regel über ein gesamtes Unternehmen und integriert zahlreiche Abteilungen, Einsatzbereiche und Vorgänge in einem einzigen Softwaresystem. Die Anwendung muss hohe Dienstqualitätsanforderungen in Bezug auf die Leistung, Verfügbarkeit, Sicherheit, Zweckmäßigkeit und Skalierbarkeit erfüllen.

Verteilte Unternehmensanwendungen erfordern eine zugrundeliegende Infrastruktur von *Diensten*, die den verteilten Komponenten die Kommunikation untereinander, die Koordination der Arbeit, die Implementierung eines sicheren Zugriffs usw. ermöglichen. Diese Infrastrukturdienste werden wiederum von einer Hardwareumgebung aus Computern und Netzwerkverbindungen unterstützt. Diese Umgebung enthält die Hardware-Architekturen SPARC® und x86 (Intel und AMD).

In der folgenden Abbildung ist das allgemeine Ebenenschema dargestellt. Java Enterprise System stellt überwiegend die Ebene der verteilten Infrastrukturdienste bereit, die in *Abbildung 1-1* dargestellt ist. Dienste von Java Enterprise System umfassen jedoch auch einige Dienste der Anwendungsebene, auf die Endbenutzer zugreifen können.

**Abbildung 1-1** Erforderliche Unterstützung für verteilte Unternehmensanwendungen

Zu den von Java Enterprise System bereitgestellten Diensten gehören:

- **Zugangsdienste:** Über diese Dienste können Mitarbeiter, Telecommuter, Knowledge Worker, Geschäftspartner, Lieferanten und Kunden von innerhalb und außerhalb des Unternehmensnetzwerks auf die Unternehmensressourcen zugreifen. Diese Dienste bieten Benutzer-Communities jederzeit und von beliebigen Standorten Zugriff und sorgen für personalisierte Integration, Aggregation, Sicherheit, mobilen Zugriff und Suchfunktionen.
- **Kommunikations- und Zusammenarbeitsdienste:** Diese Dienste ermöglichen den sicheren Austausch von Informationen innerhalb verschiedener Benutzer-Communities. Im Kontext der Unternehmensumgebung des jeweiligen Benutzers stehen spezifische Funktionen, wie beispielsweise Messaging, Zusammenarbeit in Echtzeit sowie Instant Messaging und Konferenz- oder Kalenderplanungsfunktionen zur Verfügung.
- **Netzwerkidentitäts- und Sicherheitsdienste:** Diese Dienste verbessern die Sicherheit und den Schutz wichtiger Unternehmensinformationen, indem sie auf globaler Basis die Durchsetzung entsprechender Zugriffssteuerungsrichtlinien über alle Communities, Anwendungen und Dienste hinweg gewährleisten. Diese Dienste verwenden ein zentrales Repository zum Speichern und Verwalten von Identitätsprofilen, Zugriffsrechten sowie von Informationen zu Anwendungen und Netzwerkressourcen.
- **Webcontainer- und Anwendungsdienste:** Diese Dienste sorgen dafür, dass verteilte Komponenten während der Laufzeit miteinander kommunizieren können und unterstützen die Entwicklung, Bereitstellung und Verwaltung von Anwendungen für eine Vielzahl von Servern, Clients und Geräte. Diese Dienste basieren auf der Technologie der Java 2 Platform, Enterprise Edition (J2EE™).

Das Java Enterprise System bietet außerdem Dienste, mit denen die Verfügbarkeit, Skalierbarkeit, Zweckmäßigkeit und weitere Anwendungs- oder Systemqualitäten verbessert werden. Zu den von Java Enterprise System bereitgestellten Dienstqualitätsfunktionen gehören:

- **Verfügbarkeitsdienste:** Diese Dienste sorgen für eine nahezu kontinuierliche Verfügbarkeit und bieten die Anwendungs- und Infrastrukturkomponenten, die diese unterstützen.
- **Zugriffsdienste:** Diese Dienste bieten einen internet- oder browserbasierten Zugriff auf Java Enterprise System-Dienste.
- **Verwaltungsdienste:** Diese Dienste unterstützen die Wartung und Leistungsregulierung der von Java Enterprise System unterstützten Anwendungen.

Sie können einen oder mehrere Java Enterprise System-Dienste bereitstellen, von denen jeder mehrere Java Enterprise System-Komponenten enthalten kann.

# Java Enterprise System-Komponenten

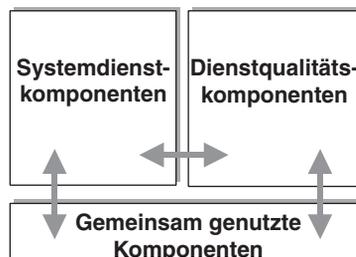
Java Enterprise System stellt eine Integration verschiedener unabhängiger Softwareprodukte in einem einzigen Softwaresystem dar. Die Komponenten dieses Systems wurden zusammen getestet, um volle Interoperabilität zu gewährleisten. Die Integration der Komponenten wird durch eine Reihe von Funktionen auf Systemebene ermöglicht:

- Alle Komponenten werden mit einem Satz gemeinsam genutzter Bibliotheken synchronisiert.
- Alle Java Enterprise System-Komponenten werden mithilfe eines einzigen Installationsprogramms installiert.
- Alle Java Enterprise System-Komponenten können gemeinsam ein integriertes Benutzeridentitäts- und Sicherheitsverwaltungssystem nutzen.

Diese Funktionen werden in den nachfolgenden Kapiteln dieses Buchs beschrieben. In diesem Abschnitt werden hauptsächlich die verschiedenen in Java Enterprise System integrierten Komponenten vorgestellt. Diese *Systemkomponenten* können, wie in der folgenden Abbildung dargestellt, in drei Hauptkategorien unterteilt werden:

- **Systemdienstkomponenten:** Diese Komponenten bieten die wesentlichen Java Enterprise System-Infrastrukturdienste für verteilte Unternehmensanwendungen.
- **Dienstqualitätskomponenten:** Diese Komponenten verbessern die Verfügbarkeit, Sicherheit, Skalierbarkeit, Zweckmäßigkeit sowie weitere Qualitäten von Systemdienstkomponenten und verteilten Anwendungskomponenten.
- **Gemeinsam genutzte Komponenten:** Diese Komponenten bilden die Umgebung, in der viele Systemdienst- und Dienstqualitätskomponenten laufen.

**Abbildung 1-2** Kategorien von Java Enterprise System-Komponenten



# Systemdienstkomponenten

Einige Java Enterprise System-Komponenten bilden die Hauptdienste für die Unterstützung verteilter Softwarelösungen. Zu diesen *Systemdiensten* gehören Zugangsdienste, Kommunikations- und Zusammenarbeitsdienste, Identitäts- und Sicherheitsdienste, Webcontainerdienste und J2EE-Anwendungsdienste.

Die folgende Tabelle enthält eine kurze Beschreibung der *Systemdienstkomponenten*, die diese verteilten Dienste bereitstellen und der von diesen angebotenen Dienste. Jede Systemdienstkomponente ist ein Servervorgang mit mehreren Threads, der eine Vielzahl von Clients unterstützen kann. Weitere Details zu den einzelnen Komponenten finden Sie unter „[Beschreibungen der Systemdienstkomponenten](#)“ auf Seite 82.

**Tabelle 1-1** Java ES-Systemdienstkomponenten

Komponente	Bereitgestellte Systemdienste
Sun Java System Access Manager	Stellt Dienste für die Zugriffsverwaltung und die digitale Identitätsverwaltung bereit. Zu den Zugriffsverwaltungsdiensten gehören die Authentifizierung (einschließlich Single Sign-On) und die rollenbasierte Autorisierung für den Zugriff auf Anwendungen und/oder Dienste. Die Verwaltungsdienste umfassen die zentralisierte Verwaltung einzelner Benutzerkonten, Rollen, Gruppen und Richtlinien.
Sun Java System Application Server	Stellt J2EE-Containerdienste für Enterprise JavaBeans™-Komponenten (EJB-Komponenten) wie Sitzungs-Beans, Einheiten-Beans und nachrichtengesteuerte Beans bereit. Der Container bietet die Infrastrukturdienste, die für die Interaktion von eng miteinander verknüpften verteilten Komponenten erforderlich sind, und macht Application Server so zu einer Plattform für die Entwicklung und Ausführung von E-Commerce-Anwendungen und Webdiensten. Der Application Server stellt darüber hinaus Webcontainerdienste bereit.
Sun Java System Calendar Server	Dieser Dienst bietet Kalender- und Planungsdienste für Endbenutzer und Endbenutzergruppen. Der Calendar Server enthält außerdem einen browserbasierten Client, der mit dem Server interagiert.
Sun Java System Directory Server	Stellt ein zentrales Repository zum Speichern und Verwalten von Intranet- und Internetinformationen, wie Identitätsprofile (Mitarbeiter, Kunden, Lieferanten usw.), Benutzeranmeldeinformationen (öffentliche Schlüsselzertifikate, Passwörter und PIN-Nummern), Zugriffsrechte sowie Anwendungs- und Netzwerkressourceninformationen bereit.
Sun Java System Instant Messaging	Ermöglicht die sichere Echtzeit-Kommunikation zwischen Endbenutzern, beispielsweise mittels Instant Messaging (Chat), Konferenzen, Warnungen, Neuigkeiten, Polling und Dateiübertragung. Dieser Dienst enthält einen Präsenz-Manager, der den Benutzern mitteilt, wer zurzeit online ist, sowie einen browserbasierten Client, der mit dem Server interagiert.

**Tabelle 1-1** Java ES-Systemdienstkomponenten (*Fortsetzung*)

<b>Komponente</b>	<b>Bereitgestellte Systemdienste</b>
Sun Java System Message Queue	Bietet zuverlässigen, asynchronen Nachrichtenaustausch zwischen lose miteinander verknüpften verteilten Komponenten und Anwendungen. Message Queue implementiert die Java Message Service (JMS) API-Spezifikation und liefert Unternehmensfunktionen wie Sicherheit, Skalierbarkeit und Remoteverwaltung.
Sun Java System Messaging Server	Bietet sicheres, verlässliches, Store-and-Forward Messaging für große Kapazitäten, das E-Mail, Fax, Pager, Sprache und Video unterstützt. Messaging Server kann gleichzeitig auf mehrere Message-Speicher zugreifen und bietet die Filterung von Inhalten, um unerwünschte E-Mails abzuweisen und Virusattacken vorzubeugen.
Sun Java System Portal Server	Stellt browserbasierten Clients, die auf Geschäftsanwendungen oder -dienste zugreifen, wichtige Zugangsdienste wie die Aggregation und Personalisierung von Inhalten bereit. Portal Server bietet außerdem eine konfigurierbare Suchmaschine.
Sun Java System Web Server	Bietet J2EE™-Webcontainerdienste für Java-Webkomponenten, wie die Komponenten Java Servlet und JavaServer Pages™ (JSP™). Web Server unterstützt darüber hinaus weitere Web-Anwendungstechnologien für die Auslieferung statischer und dynamischer Web-Inhalte, wie CGI-Skripts und SUN Java System Active Server Pages.

## Dienstqualitätskomponenten

Neben den in der [Tabelle 1-1](#) aufgeführten Systemdienstkomponenten enthält das Java Enterprise System einige Komponenten, die die Qualität der von Systemdienstkomponenten oder von kundenspezifisch entwickelte Anwendungskomponenten angebotenen Dienste verbessern. Diese *Dienstqualitätskomponenten* fallen in folgende Kategorien:

- Verfügbarkeitskomponenten
- Zugriffskomponenten
- Verwaltungskomponenten

## Verfügbarkeitskomponenten

Verfügbarkeitskomponenten sorgen für eine nahezu kontinuierliche Systembetriebszeit von Systemdienstkomponenten und Anwendungskomponenten. Die folgende Tabelle enthält die in Java Enterprise System enthaltenen Komponenten und die von diesen angebotenen Dienste. Weitere Details zu den einzelnen Komponenten finden Sie unter [„Beschreibung der Verfügbarkeitskomponenten“ auf Seite 87](#).

**Tabelle 1-2** Java ES-Verfügbarkeitskomponenten

Komponente	Bereitgestellte Verfügbarkeitsdienste
Sun Cluster	Bietet Hochverfügbarkeits- und Skalierbarkeitsdienste für Java Enterprise System und für die Anwendungen, die basierend auf der Java Enterprise System-Infrastruktur ausgeführt werden, sowie für die Hardwareumgebung, in der beide bereitgestellt werden.
High Availability Session Store	Bietet einen Datenspeicher, der Anwendungsdaten, insbesondere Sitzungsstatusdaten, auch im Fehlerfall verfügbar macht.

## Zugriffskomponenten

Zugriffskomponenten bieten Front-End-Zugriff auf Systemdienste und sorgen für einen sicheren Zugriff über Internetstandorte, die sich außerhalb der Firewall des Unternehmens befinden. Neben diesen Zugriffsmöglichkeiten bieten viele auch eine Routing-Funktion. Die folgende Tabelle zeigt die in Java Enterprise System enthaltenen Komponenten und die von diesen angebotenen Dienste. Weitere Details zu den einzelnen Komponenten finden Sie unter [„Beschreibung der Zugriffskomponenten“ auf Seite 89](#).

**Tabelle 1-3** Java ES-Zugriffskomponenten

Komponente	Bereitgestellte Zugriffsdienste
Sun Java System Directory Proxy Server	Bietet Sicherheitsdienste für Directory Server von außerhalb der Unternehmens-Firewall. Directory Proxy Server bietet eine verbesserte Kontrolle des Verzeichniszugriffs, Schemakompatibilität, Attributfilterung und Routing für mehrere Directory Server-Instanzen.
Sun Java System Portal Server Secure Remote Access	Ermöglicht außerhalb einer Unternehmens-Firewall den sicheren Internetzugriff auf den Inhalt und die Dienste von Portal Server, einschließlich interne Portale und Internetanwendungen.

**Tabelle 1-3** Java ES-Zugriffskomponenten (*Fortsetzung*)

<b>Komponente</b>	<b>Bereitgestellte Zugriffsdienste</b>
Sun Java System Communications Express	Bietet webbasierten Zugriff auf Messaging Server, Calendar Server und Directory Server, je nach Konfiguration.
Sun Java System Connector für Microsoft Outlook	Bietet für Desktop-Clients, die Microsoft Outlook verwenden, eine Schnittstelle zu Messaging Server und Calendar Server.

## Verwaltungskomponenten

Verwaltungskomponenten bieten Verwaltungsfunktionen für Systemdienste, wie Konfiguration und Überwachung. Die folgende Tabelle enthält die in Java Enterprise System enthaltenen Verwaltungskomponenten und die von diesen angebotenen Dienste. Weitere Details zu den einzelnen Komponenten finden Sie unter „[Beschreibung der Verwaltungskomponenten](#)“ auf Seite 91.

**Tabelle 1-4** Java ES-Verwaltungskomponenten

<b>Komponente</b>	<b>Bereitgestellte Verwaltungsdienste</b>
Sun Java System Administration Server (und Konsole).	Bietet ein grafisches Verwaltungs-Tool mit dem Sie Directory Server und Messaging Server konfigurieren und verwalten können.
Sun Java System Directory Preparation Tool	Bietet ein Skript für die Konfiguration von Directory Server mit dem Schema, das benötigt wird, damit Benutzer auf Messaging Server und Calendar Server zugreifen können.
Sun Java System Delegated Administrator	Bietet Tools für die Befehlszeile und die Benutzeroberfläche, mit denen Benutzer Einträge in Directory Server mit den von Messaging Server und Calendar Server benötigten Benutzerattributen auffüllen können.
Sun Remote Services Net Connect	Bietet die Möglichkeit der Remote-Überwachung.

## Gemeinsam genutzte Komponenten

Java Enterprise System enthält einige lokal installierte gemeinsam genutzte Bibliotheken, von denen viele Systemdienstkomponenten und Dienstqualitätskomponenten abhängen. Java Enterprise System *Gemeinsam genutzte Komponenten* bieten lokale Dienste für Java Enterprise System-Komponenten, die auf demselben Hostcomputer laufen.

Gemeinsam genutzte Komponenten werden oft eingesetzt, um die Portierbarkeit zwischen unterschiedlichen Betriebssystemen zu gewährleisten. Beispiele für gemeinsam genutzte Komponenten von Java Enterprise System sind u. a.: Java 2 Platform, Standard Edition (J2SE™-Plattform), Netscape Portable Runtime (NSPR), Network Security Services (NSS), Network Security Services for Java (JSS) usw. Eine vollständige Liste finden Sie unter „*Gemeinsam genutzte Komponenten*“ auf [Seite 92](#).

# Arbeiten mit Java Enterprise System

Beim Erstellen von auf Java Enterprise System-Software basierenden Unternehmenslösungen fallen einige Standardaufgaben an. Diese Aufgaben sind, abhängig Startpunkt des Einsatzes von Java Enterprise System und von der Art der Lösung, die Sie erstellen und bereitstellen wollen, unterschiedlich aufwändig und unterschiedlich schwierig.

Dieser Abschnitt behandelt zwei Aspekte des Arbeitens mit Java Enterprise System: Den Java Enterprise System-Lösungslebenszyklus und die verschiedenen Einsatzszenarios, die in der Regel bestehen.

## Java Enterprise System Lösungslebenszyklus

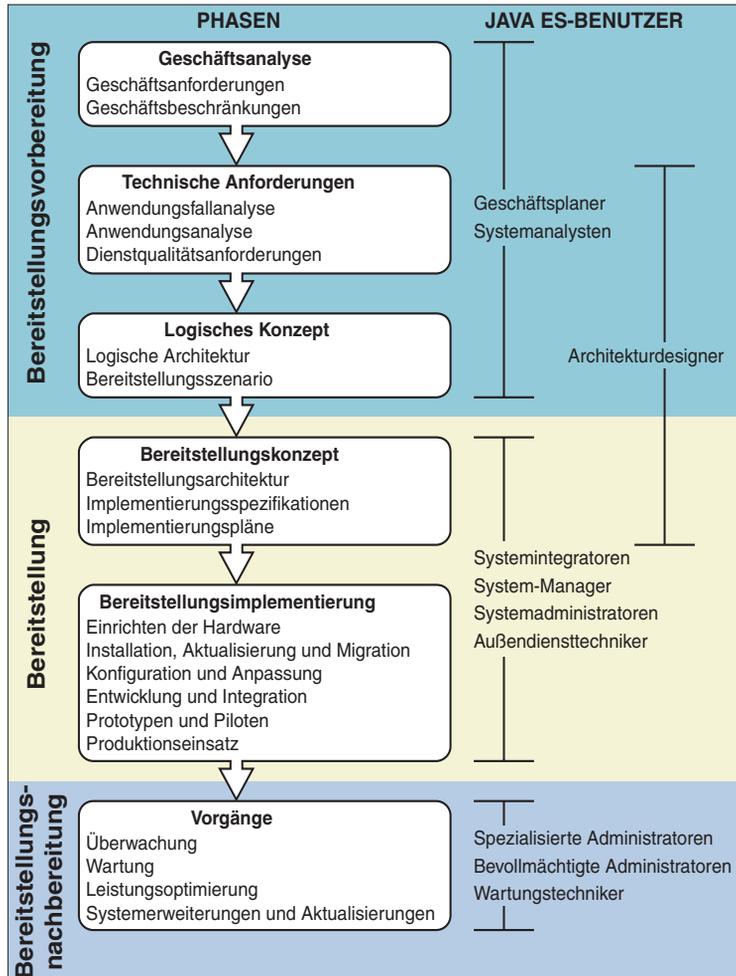
Die Aufgaben, die beim Erstellen von auf Java Enterprise System-Software basierenden Unternehmenslösungen anfallen, können in mehrere Phasen unterteilt werden, die in [Abbildung 1-3 auf Seite 28](#) dargestellt sind. Die Abbildung zeigt außerdem, welche Kategorien von Java Enterprise System-Benutzern diese Aufgaben normalerweise durchführen.

Die in [Abbildung 1-3](#) gezeigten Lebenszyklusphasen können in folgende allgemeine Gruppen unterteilt werden:

- **Bereitstellungsvorbereitung.** In dieser Phase wird eine Unternehmensanforderung in ein Bereitstellungsszenario übersetzt: Eine logische Architektur und einen Satz von Dienstqualitätsanforderungen. Das Bereitstellungsszenario dient als Spezifikation für die Konzeption einer Bereitstellungsarchitektur.
- **Bereitstellung:** In dieser Phase wird ein Bereitstellungsszenario in eine Bereitstellungsarchitektur übersetzt. Diese Architektur kann als Grundlage für die Genehmigung und Budgetierung des Projekts verwendet werden. Diese Bereitstellungsarchitektur ist auch die Grundlage für eine Implementierungsspezifikation, die die erforderlichen Details für die Bereitstellung (Erstellung, Testen und Einsatz) einer Softwarelösung in einer Produktionsumgebung enthält.
- **Bereitstellungsnachbereitung:** In der Einsatzphase läuft eine bereitgestellte Lösung unter Produktionsbedingungen und wird hinsichtlich der Leistung überwacht und optimiert. Die bereitgestellte Lösung wird bei Bedarf mit neuen Funktionen aktualisiert.

Der Lösungslebenszyklus und die in jeder Phase anfallenden Aufgaben (siehe [Abbildung 1-3](#)) werden eingehender vorgestellt im [Kapitel 4](#), „Aufgaben des Lösungslebenszyklus von Java Enterprise System“.

**Abbildung 1-3** Phasen des Lebenszyklus einer Lösung und Benutzerkategorien



**Abbildung 1-3** zeigt die Java Enterprise System-Benutzer, die üblicherweise die für die Lebenszyklusphase angegebenen Aufgaben durchführen. Wenn Sie mit Java Enterprise System arbeiten, sollte Ihre Funktion einer oder mehreren der Benutzerkategorien entsprechen, die in **Abbildung 1-3** aufgeführt sind. Die folgende Tabelle beschreibt die Kenntnisse und den Hintergrund der einzelnen Benutzerkategorien.

**Tabelle 1-5** Java ES-Benutzerkategorien für Aufgaben des Lebenszyklus

<b>Benutzer</b>	<b>Fähigkeiten und Kenntnisse</b>	<b>Phasen</b>
Geschäftsplaner Systemanalyst	Eher allgemeines, anstatt tiefer gehendes technisches Wissen. Verständnis der strategischen Orientierung des Unternehmens. Kenntnis der Geschäftsprozesse, -ziele und -anforderungen.	Geschäftsanalyse Technische Anforderungen Logisches Konzept
Architekt	Sehr technisch orientiert. Breite Kenntnis der Bereitstellungsarchitekturen. Kenntnis der neuesten Technologien. Verständnis der Geschäftsanforderungen und -beschränkungen.	Logisches Konzept Bereitstellungskonzept
Systemintegrator Außendiensttechniker Systemadministrator System-Manager	Sehr technisch orientiert. Tiefgehende Kenntnis der IT-Umgebungen. Erfahrung mit der Implementierung verteilter Softwarelösungen. Kenntnis der Netzwerkarchitektur, Protokolle, Geräte und Sicherheit. Kenntnis der Skript- und Programmiersprachen.	Bereitstellungskonzept Bereitstellungs- implementierung
Spezialisierte Systemadministrator Bevollmächtigter Administrator Wartungstechniker	Spezielle technische Kenntnisse oder Produktkenntnisse. Gute Kenntnis der Hardware, Plattformen, Verzeichnisse und Datenbanken. Geschult in der Überwachung, Fehlerbehebung und Aufrüstung von Software. Kenntnis der Systemverwaltung für Betriebssystemplattformen.	Vorgänge

# Java Enterprise System Einführungsszenarien

Der Unternehmensbedarf, der zur Einführung von Java Enterprise System führt, ist sehr unterschiedlich. Das höchste Ziel nahezu jeder Java Enterprise System-Bereitstellung passt allerdings zu einem wie folgt aussehenden *Einführungsszenario*:

- **Neues System:** Sie besitzen noch kein Softwaresystem und beginnen mit der Bereitstellung der Java Enterprise System-Software, um eine neue Geschäftslösung zu unterstützen.
- **Verbesserung:** Sie besitzen eine IT-Infrastruktur und ersetzen einige oder alle Teile Ihres Systems durch Java Enterprise System-Software. Üblicherweise werden Systeme oder Subsysteme ersetzt, weil deren Erhaltung zu kompliziert, zu eingeschränkt oder zu teuer wäre. Sie können beispielsweise bessere Sicherheit, höhere Verfügbarkeit, mehr Skalierbarkeit, mehr Flexibilität, weniger Komplexität, zusätzliche Funktionen (wie Single Sign-On) oder eine bessere Nutzung von IT-Ressourcen benötigen. Anders ausgedrückt: Sie wünschen sich eine bessere Investitionsrentabilität, als es Ihr vorhandenes System bietet.
- **Erweiterung:** Sie besitzen eine IT-Infrastruktur und beginnen mit der Bereitstellung von Java Enterprise System-Software, die in Ihrem System nicht vorhanden ist. Üblicherweise werden Softwaresysteme auf diese Art erweitert, weil neue Geschäftsanforderungen erfüllt werden müssen. Sie benötigen gegebenenfalls neue Funktionen, wie personalisierte Aggregation vorhandener Dienste über ein Java Enterprise System-Portal oder Java-Authentifizierung und -Autorisierung für vorhandene Dienste.
- **Aufrüstung:** Sie besitzen eine IT-Infrastruktur, die aus einer früheren Version von Java Enterprise System oder aus früheren Sun-Produkten besteht, und nehmen eine Aufrüstung auf die aktuellste Version der Java Enterprise System-Komponenten vor.

Jedes Einführungsszenario bietet eigene Aspekte und Herausforderungen. Unabhängig davon, welches Szenario Ihre Situation kennzeichnet, der in [Abbildung 1-3](#) gezeigte Prozess des Lösungslebenszyklus zutrifft. Abhängig von Ihrem Einführungsszenario sind die in der Lebenszyklusphase zu lösenden Probleme und die zu investierenden Ressourcen jedoch unterschiedlich.

Die folgenden Punkte gelten abgestuft für die Einföhrungsszenarien:

- **Migration:** Die Verbesserung oder Aufrüstung der vorhandenen Infrastruktur durch neue Software macht häufig die Migration von Daten aus vorhandenen in neue Systeme notwendig. Bei den Daten kann es sich um Konfigurations-, Benutzer- oder Anwendungsinformationen handeln. Gegebenenfalls müssen Sie aufgrund neuer Programmierschnittstellen außerdem die Geschäfts- oder Darstellungslogik migrieren.
- **Integration:** Das Hinzufügen neuer Software zu einem vorhandenen System oder das Ersetzen von Software-Subsystemen führt oft dazu, dass Sie neue Softwarekomponenten in die verbleibenden Subsysteme integrieren müssen. Zur Integration kann die Entwicklung neuer Schnittstellenebenen, der Einsatz von J2EE Connectoren oder Ressourcen-Adaptern, die Neukonfiguration vorhandener Komponenten sowie die Implementierung von Datentransformationsschemata gehören.
- **Schulung:** Nahezu jede Veränderung der Infrastruktur führt zu Änderungen der IT-Verfahren und der notwendigen Kenntnisse. Ihre IT-Abteilung benötigt einen angemessenen Zeitraum, um die Kenntnisse für die neuen Java Enterprise System-Technologien zu erlangen oder die vorhandenen Kenntnisse weiterzuvermitteln.
- **Hardware:** Wenn Sie ein vorhandenes System oder Subsystem ersetzen, können es die Unternehmensbedingungen erforderlich machen, dass vorhandene Hardware weiterverwendet wird. Abhängig von Ihrem Einföhrungsszenario können die Hardwareressourcen zu einem wichtigen Faktor werden.

Die folgende Tabelle fasst die Art der Punkte zusammen, die bei den einzelnen Einföhrungsszenarien von Java Enterprise System von Bedeutung sind.

**Tabelle 1-6** Aspekte unterschiedlicher Java ES-Einföhrungsszeanrien

Einföhrungsszenarien	Migration	Integration	Schulung	Hardware
Neues System	Nicht zutreffend	Relativ einfache Integration neuer Komponenten	Normalerweise ein wichtiger Aspekt	Ausgleich zwischen Ausrüstungskosten und Arbeitskosten.*
Verbesserung	Kann ein bedeutender Aspekt sein	Neue Komponenten müssen in vorhandene Systeme integriert werden	Kann ein wichtiger Aspekt sein	Kann wegen vorhandener Anlagen zu signifikanten Einschränkungen führen

**Tabelle 1-6** Aspekte unterschiedlicher Java ES-Einführungsszeanrien

Einführungsszenarien	Migration	Integration	Schulung	Hardware
Erweiterung	Üblicherweise kein Aspekt	Gegebenenfalls müssen neue Komponenten in vorhandene Systeme integriert werden	Gegebenenfalls ein wichtiger Aspekt	Macht üblicherweise neue Hardware notwendig, mit denselben Abwägungen wie bei einem neuen System
Aufrüstung	Kann ein wichtiger Aspekt sein	Relativ einfache Integration aufgerüsteter Komponenten	Relativ unwesentlicher Aspekt	Relativ unwesentlicher Aspekt

\* Der Einsatz einiger leistungsstarker Computer erhöht im Allgemeinen die Anlagekosten, wenn weniger IT-Ressourcen benötigt werden. Der Einsatz vieler kleiner Computer senkt im Allgemeinen die Anlagekosten, wenn mehr IT-Ressourcen benötigt werden.

## In diesem Kapitel enthaltene Schlüsselbegriffe

Dieser Abschnitt erläutert in diesem Kapitel verwendete wichtige technische Begriffe, wobei der Schwerpunkt darauf liegt zu verdeutlichen, welche Beziehung zwischen diesen Begriffen im Java Enterprise System-Kontext vorliegt.

**Einführungsszenario** Der wesentliche Grund für die Bereitstellung der Java Enterprise System-Software, zusammen mit den Merkmalen des Softwaresystems, mit dem Sie beginnen und dem von Ihnen gewünschten Ziel. Es gibt vier grundlegende Einführungsszenarien für Java Enterprise System: Neues System, Erneuerung, Erweiterung und Aufrüstung.

**Komponente** Eine Softwarelogikeinheit, die zum Aufbau von verteilten Anwendungen verwendet wird. Bei einer Komponente kann es sich um eine der im Java Enterprise System enthaltenen *Systemkomponenten* oder um eine kundenspezifisch entwickelte *Anwendungskomponente* handeln. Eine Anwendungskomponente entspricht in der Regel einem Modell verteilter Komponenten (wie CORBA und die J2EE™-Plattform). Sie führt bestimmte Datenverarbeitungsfunktionen durch. Einzeln oder kombiniert bieten diese Komponenten *Geschäftsdienste* und können zu *Webdiensten* zusammengefasst werden.

**Verteilte Unternehmensanwendung** Eine Anwendung, deren Logik sich über eine Netzwerk- oder Internetumgebung hinweg erstreckt (der Verteilungsaspekt) und deren Umfang und Größe die Anforderungen einer Produktionsumgebung oder eines Dienstansbieters erfüllen (der Unternehmensaspekt).

**Endbenutzer** Ein Mitarbeiter, der eine verteilte Anwendung nutzt, häufig über eine grafische Benutzeroberfläche, wie die Oberfläche eines Internetbrowsers oder eines mobilen Geräts. Die Anzahl der gleichzeitig von einer Anwendung unterstützten Endbenutzer bildet einen wichtigen Faktor für die *Bereitstellungsarchitektur* der Anwendung.

**Dienst** Eine Softwarefunktion, die für einen oder mehrere *Clients* ausgeführt wird. Diese Funktion kann sich auf einer sehr niedrigen oder einer sehr hohen Ebene befinden, wie beispielsweise eine Kreditüberprüfung durch einen *Geschäftsdienst*. Ein auf hoher Ebene liegender Dienst kann aus einer ganzen Reihe einzelner Dienste bestehen. Die Dienste können lokal (für lokale Clients verfügbar) oder verteilt (für Remote-Clients verfügbar) sein.

**Dienstqualitätskomponente** Eine der in Java Enterprise System enthaltenen Arten von *Systemkomponenten*. Wartungskomponenten, die Zugriffs- und Verwaltungskomponenten enthalten, bieten Unterstützung für *Systemdienstkomponenten*.

**Gemeinsam genutzte Komponente** Eine der in Java Enterprise System enthaltenen Arten von *Systemkomponenten*. Gemeinsam genutzte Komponenten, in der Regel Bibliotheken, bieten lokale Dienste für andere Systemkomponenten. Im Gegensatz dazu stellt eine *Systemdienstkomponente* verteilte Infrastrukturdienste für andere Systemkomponenten (oder für *Anwendungskomponenten*) bereit.

**Systemkomponente** Alle Softwarepakete oder Gruppen von Paketen, die Teil von Java Enterprise System sind und vom Java Enterprise System-Installationsprogramm installiert werden. Es gibt verschiedene Arten von Systemkomponenten: *Systemdienstkomponenten*, die verteilte Infrastruktur-*Dienste* bieten, *Dienstqualitätskomponenten*, die die Systemdienstkomponenten durch Zugriffs- und Verwaltungsdienste unterstützen, und *Gemeinsam genutzte Komponenten*, die lokale Dienste für andere Systemkomponenten bieten.

**Systemdienst** Einer oder mehrere verteilte *Dienste*, die die von einem Java Enterprise System bereitgestellten einzigartigen Funktionen definieren. Systemdienste erfordern in der Regel die Unterstützung durch eine Reihe von *Dienstqualitätskomponenten*, *Gemeinsam genutzte Komponenten* oder mehrere von beiden.

**Systemdienstkomponente** Eine der in Java Enterprise System enthaltenen Arten von *Systemkomponenten*. Systemdienstkomponenten bilden die wesentlichen Infrastrukturdienste von Java Enterprise System: Zugangsdienste, Kommunikations- und Zusammenarbeitsdienste, Identitäts- und Sicherheitsdienste, Web- und Anwendungsdienste sowie Verfügbarkeitsdienste.

In diesem Kapitel enthaltene Schlüsselbegriffe

# Lösungsarchitekturen von Java Enterprise System

Dieses Kapitel enthält eine Übersicht der Architekturkonzepte, auf denen die Java Enterprise System-Lösungen (Java ES-Lösungen) basieren. In diesem Kapitel wird gezeigt, wie Java ES-Komponenten (Systemdienstkomponenten und Dienstqualitätskomponenten) eingesetzt werden, um verteilte Unternehmenslösungen zu unterstützen.

*Architekturen* von Java ES-Lösungen bieten zwei Aspekte: Eine *logische Architektur* und eine *Bereitstellungsarchitektur*. Die logische Architektur stellt die Interaktionen zwischen den logischen Modulblöcken (den Softwarekomponenten) einer Lösung dar. Die Bereitstellungsarchitektur stellt die Zuordnung der logischen Architektur zu einer physischen Computerumgebung dar. Java ES-Komponenten spielen sowohl in der logischen als auch in der Bereitstellungsarchitektur eine wichtige Rolle.

In diesem Kapitel wird ein Architekturframework für die Konzeption der Architektur von Java ES-Lösungen beschrieben, das anschließend in einer auf dem Architekturframework aufbauenden Beispiellösung dargestellt wird.

Das Kapitel behandelt folgende Themen:

- „Java Enterprise System-Architekturframework“ auf Seite 36
- „Beispiel für Java Enterprise System-Lösungsarchitektur“ auf Seite 53
- „In diesem Kapitel enthaltene Schlüsselbegriffe“ auf Seite 57

# Java Enterprise System-Architekturframework

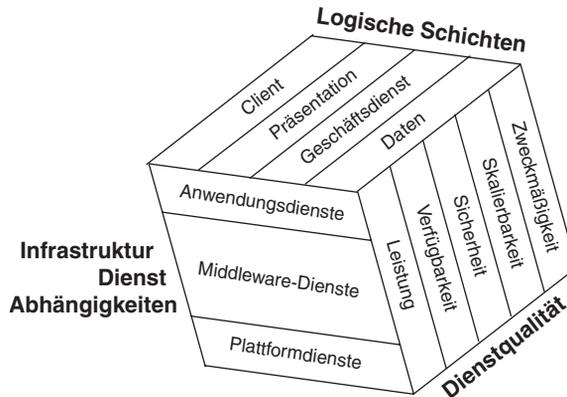
Java Enterprise System-Komponenten unterstützen die Bereitstellung verteilter Softwarelösungen für die Anforderungen von Unternehmen.

Damit die benötigte Funktionalität auf der von den Geschäftsanforderungen vorgegebenen Ebene der Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit erreicht wird, müssen solche Softwarelösungen korrekt entworfen werden.

Bei der Konzeption verteilter Softwarelösungen in Unternehmensstärke sind verschiedene Architekturdimensionen zu berücksichtigen. Diese Dimensionen repräsentieren unterschiedliche Perspektiven, aus denen die Interaktionen der vielen, solche Systeme bildenden Softwarekomponenten betrachtet werden. Insbesondere bei der Konzeption verteilter Systeme sind die drei folgenden Architekturdimensionen zu berücksichtigen:

- **Infrastrukturdienstabhängigkeiten:** Diese Dimension stellt die Rolle der Systemdienstkomponenten (siehe „[Systemdienstkomponenten](#)“ auf Seite 22) bei der Unterstützung verteilter Lösung in den Vordergrund.
- **Logische Schichten:** Diese Dimension stellt die logische und physische Unabhängigkeit von Lösungskomponenten in den Vordergrund, die in einer Netzwerk- oder Internetumgebung bereitgestellt werden sollen.
- **Dienstqualität:** Diese Dimension stellt dar, wie Dienstqualitätsanforderungen, wie Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit, erreicht werden, wobei die Rolle der Dienstqualitätskomponenten berücksichtigt wird (siehe „[Dienstqualitätskomponenten](#)“ auf Seite 23).

In der folgenden Abbildung sind diese drei Dimensionen der Lösungsarchitektur dargestellt.

**Abbildung 2-1** Dimensionen der Java ES-Lösungsarchitektur

Diese drei Dimensionen bilden zusammen ein einziges Framework, das die Beziehungen zwischen den Softwarekomponenten (*Anwendungskomponenten* und Infrastrukturkomponenten) berücksichtigt, die für das Erreichen der für eine Softwarelösung benötigten Dienstfunktionen und der Dienstqualität erforderlich sind.

In den folgenden Abschnitten werden die drei Dimensionen nacheinander beschrieben. Anschließend wird eine Synthese der drei Dimensionen in einem zusammengeführten Framework vorgenommen.

## Dimension 1: Infrastrukturdienstabhängigkeiten

Die interagierenden Softwarekomponenten verteilter Unternehmensanwendungen setzen eine Reihe von Basisinfrastrukturdiensten voraus, die den verteilten Komponenten die Kommunikation untereinander, die Koordination ihrer Arbeit, die Implementierung eines sicheren Zugriffs usw. ermöglichen. In diesem Abschnitt wird die Schlüsselrolle einiger Java ES-Komponenten bei der Bereitstellung dieser Infrastrukturdienste erläutert.

### Infrastrukturdienstebenen

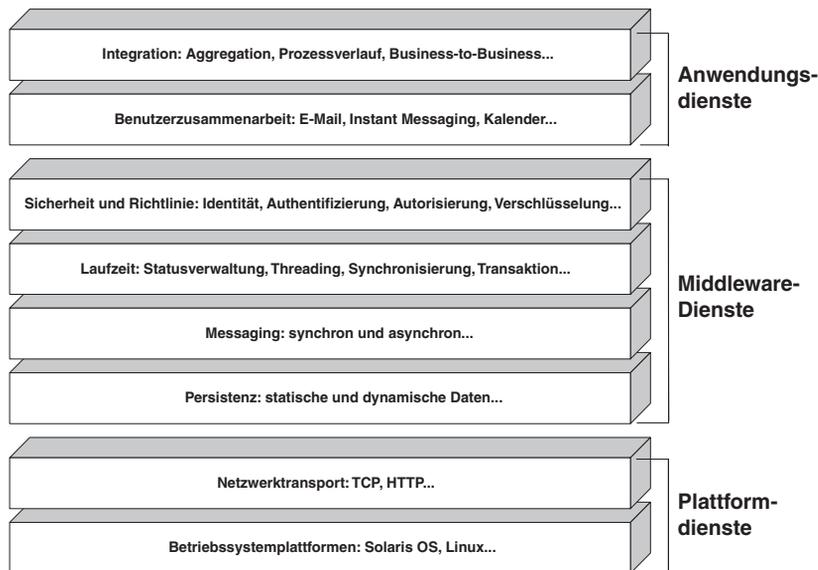
Unabhängig davon, ob ein verteiltes Softwaresystem überwiegend aus selbst entwickelten oder fertigen Java ES-Komponenten besteht, müssen Sie bei der Konzeption einige Infrastrukturdienste berücksichtigen. Diese Dienste arbeiten auf vielen Ebenen.

[Abbildung 2-2 auf Seite 38](#) illustriert die Dimension der Infrastrukturdienstabhängigkeiten in der Lösungsarchitektur. Die in dieser Abbildung dargestellten Ebenen sind eine vergrößerte Ansicht der in [Abbildung 1-1 auf Seite 19](#) gezeigten Infrastrukturdienstebene.

Die Hierarchie der in [Abbildung 2-2](#) dargestellten Dienste und die Abhängigkeiten zwischen den Diensten bilden eine wichtige Dimension der logischen Architektur der Lösung. Diese Infrastrukturdienste bieten die konzeptionelle Grundlage, anhand der die Rolle der Java ES-Systemdienstkomponenten (siehe [„Systemdienstkomponenten“ auf Seite 22](#)) verständlich wird.

Im Allgemeinen können die in [Abbildung 2-2](#) dargestellten Dienste in drei große Gruppen unterteilt werden: Plattformdienste auf niedrigster Ebene, Anwendungsdienste auf höchster Ebene und eine Gruppe von Middleware-Diensten, die ihren Namen ihrer Position zwischen den beiden anderen Gruppen verdanken.

**Abbildung 2-2** Dimension 1: Infrastrukturdienstebenen



In den folgenden Absätzen werden die unterschiedlichen Infrastrukturdienstebenen beschrieben und, sofern relevant, der Bezug zu den Artefakten der Programmiersprache Java hergestellt. Die Beschreibung der Dienstebenen erfolgt, wie in [Abbildung 2-2](#) dargestellt, von der niedrigsten Ebene bis hinauf zur höchsten Ebene:

- **Betriebssystemplattformen:** Bieten die Basisunterstützung für alle Prozesse, die auf einem Computer ausgeführt werden. Das Betriebssystem (wie Solaris™ Operating System, Linux oder Microsoft Windows) verwaltet die physischen Geräte sowie den Arbeitsspeicher, die Threads und andere Ressourcen, die zur Unterstützung der Java Virtual Machine (JVM™) erforderlich sind.
- **Netzwerktransport:** Bietet die grundlegende Netzwerkunterstützung für die Kommunikation zwischen den verteilten Anwendungskomponenten, die auf den verschiedenen Computern ausgeführt werden. Zu diesen Diensten gehört auch die Unterstützung von Protokollen wie TCP und HTTP. Andere Kommunikationsprotokolle auf höheren Ebenen (siehe Messaging-Ebene) sind von diesen grundlegenden Transportdiensten abhängig.
- **Persistenz:** Bietet Unterstützung für den Zugriff auf und die Speicherung von statischen Daten (wie Benutzer-, Verzeichnis- oder Konfigurationsinformationen) und dynamischen Anwendungsdaten (Informationen, die häufig aktualisiert werden).
- **Messaging:** Bietet Unterstützung für die synchrone und asynchrone Kommunikation zwischen den Anwendungskomponenten. Synchrones Messaging besteht im Senden und Empfangen von Nachrichten in Echtzeit und umfasst Remote-Methodenaufrufe (RMI) zwischen J2EE-Komponenten und SOAP-Interaktionen zwischen Webdiensten. Asynchrones Messaging besteht in einer Kommunikation, bei der das Senden einer Nachricht nicht davon abhängt, ob der Konsument bereit ist, diese sofort zu empfangen. Die Spezifikationen für asynchrones Messaging, beispielsweise Java Message Service (JMS) und ebXML, sorgen für garantierte Zuverlässigkeit und andere Aspekte der Messaging-Semantik.
- **Laufzeit:** Bietet die Unterstützung, die für jedes verteilte Komponentenmodell, beispielsweise J2EE- oder CORBA-Modelle, erforderlich ist. Neben dem Remote-Methodenaufruf, der für eng miteinander verknüpfte verteilte Komponenten benötigt wird, umfassen die Laufzeitdienste die Komponentenstatusverwaltung (Lebenszyklusverwaltung), die Thread-Pool-Verwaltung, die Synchronisierung (Mutex-Sperrung), Persistenzdienste, die verteilte Transaktionsüberwachung und die verteilte Ausnahmenverarbeitung. In einer J2EE-Umgebung werden diese Laufzeitdienste von EJB™-Containern, Webcontainern und nachrichtengesteuerten Bean-Containern (MDB-Containern) auf einem Anwendungs- oder Webserver bereitgestellt.

- **Sicherheit und Richtlinie:** Bietet Unterstützung für den sicheren Zugriff auf Anwendungsressourcen. Diese Dienste umfassen die Unterstützung für Richtlinien, die den gruppen- oder rollenbasierten Zugriff auf verteilte Ressourcen steuern, sowie *Single Sign-On*-Funktionen. Single Sign-On ermöglicht, dass die Authentifizierung eines Benutzers bei einem Dienst in einem verteilten System automatisch auf andere Dienste (J2EE-Komponenten, Geschäftsdienste und Webdienste) in diesem System angewendet wird.
- **Benutzerzusammenarbeit:** Bietet Dienste, die bei der Unterstützung der direkten Kommunikation zwischen den Benutzern und der Zusammenarbeit der Benutzer in Unternehmens- und Internetumgebungen eine wichtige Rolle spielen. Diese Dienste sind Geschäftsdienste auf Anwendungsebene, die in der Regel von eigenständigen Servern (beispielsweise von einem E-Mail- oder Kalenderserver) bereitgestellt werden.
- **Integration:** Stellt die Dienste für die Aggregation vorhandener Geschäftsdienste bereit. Bietet eine gemeinsame Schnittstelle für den Zugriff auf die Dienste, wie dies in einem Portal der Fall ist, indem die Dienste über eine Prozess-Engine integriert werden, die diese innerhalb eines Produktionsworkflows koordiniert. Die Integration kann auch in Form von Business-to-Business-Interaktionen zwischen verschiedenen Unternehmen erfolgen.

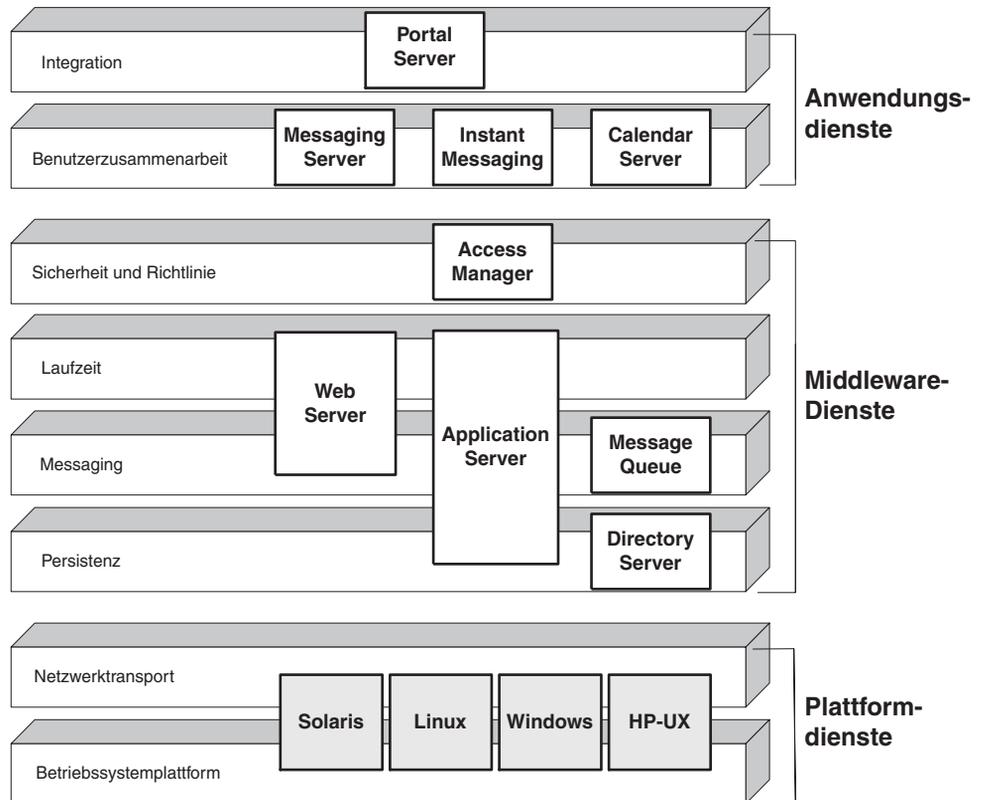
Die in [Abbildung 2-2](#) dargestellten Dienstebenen spiegeln die allgemeine gegenseitige Abhängigkeit der verschiedenen Infrastrukturdienste wider, von der untersten Ebene der Betriebssystemdienste bis hinauf zur höchsten Ebene der Anwendungs- und Integrationsdienste. Im Allgemeinen ist jeder Dienst von untergeordneten Diensten abhängig und unterstützt selbst übergeordnete Dienste.

[Abbildung 2-2](#) stellt jedoch keine strenge Ebenenschichtung der Infrastrukturdienste dar. Dienste höherer Ebenen können direkt mit Diensten niedrigerer Ebenen interagieren, ohne von Zwischenebenen abhängig zu sein. So sind beispielsweise bestimmte Laufzeitdienste direkt von den Plattformdiensten abhängig, ohne eine der dazwischen liegenden Dienstebenen zu benötigen. Zusätzlich könnten auch andere Dienstebenen, wie Überwachungs- oder Verwaltungsdienste, ebenfalls in diese Konzeptdarstellung aufgenommen werden.

## Java Enterprise System-Infrastrukturdienstkomponenten

Java ES-Komponenten implementieren die verteilten Infrastrukturdienstebenen, die in [Abbildung 2-2](#) dargestellt sind. Die Positionierung der Java ES-Systemdienstkomponenten innerhalb der einzelnen Ebenen ist in [Abbildung 2-3](#) dargestellt.

**Abbildung 2-3** Java ES Systemdienstkomponenten



**HINWEIS** Die in [Abbildung 2-3](#) dargestellten Betriebssystemplattformen sind kein formeller Bestandteil von Java Enterprise System. Sie sind jedoch darin enthalten, um die Betriebssystemplattformen darzustellen, auf denen die Java ES-Komponenten unterstützt werden.

## Java Enterprise System Infrastrukturdienstabhängigkeiten

Im Allgemeinen ist jede der in [Abbildung 2-3](#) dargestellten Java ES-Systemdienstkomponenten von in der Infrastruktur unter ihr liegenden Komponenten anhängig und unterstützt darüber liegende Komponenten. Diese Beziehung von Abhängigkeit und Unterstützung bildet einen Schlüsselfaktor bei der Konzeption logischer Architekturen.

[Tabelle 2-1](#) zeigt die spezifischen Abhängigkeiten zwischen den Java ES-Systemdienstkomponenten, die in [Abbildung 2-3](#) von oben nach unten verlaufend dargestellt sind.

**Tabelle 2-1** Beziehungen zwischen Java ES -Systemdienstkomponenten

Komponente	Ist abhängig von	Bietet Unterstützung für
Portal Server	Application Server oder Web Server Access Manager Directory Server Wenn zur Verwendung von entsprechenden Kanälen konfiguriert: Calendar Server Messaging Server Instant Messaging	
Messaging Server	Directory Server Access Manager (für Single Sign-On)	Calendar Server (für E-Mail-Benachrichtigungen) Portal Server (für den Messaging-Kanal)
Instant Messaging	Directory Server Access Manager (für Single Sign-On)	Portal Server (für den Instant Messaging-Kanal)
Calendar Server	Directory Server Messaging Server (für den E-Mail-Benachrichtigungsdienst) Access Manager (für Single Sign-On)	Portal Server (für den Kalenderkanal)
Access Manager	Application Server oder Web Server Directory Server	Portal Server Wenn konfiguriert für Single Sign-On: Calendar Server Messaging Server Instant Messaging
Application Server	Message Queue Directory Server (für verwaltete Objekte)	Portal Server Access Manager

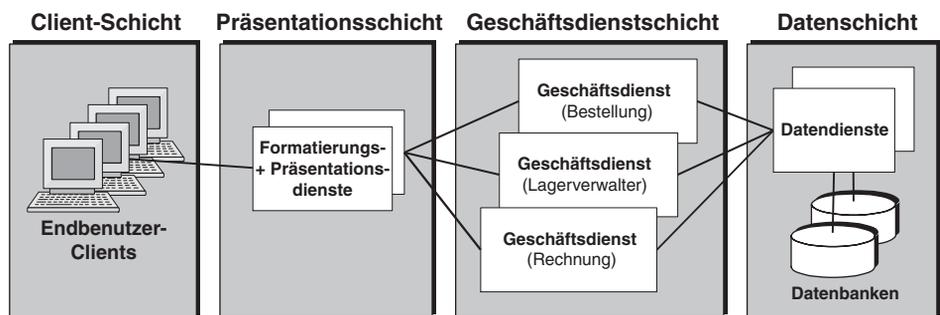
**Tabelle 2-1** Beziehungen zwischen Java ES -Systemdienstkomponenten (Fortsetzung)

Komponente	Ist abhängig von	Bietet Unterstützung für
Message Queue	Directory Server (für verwaltete Objekte)	Application Server
Web Server	Access Manager (für die Zugriffssteuerung)	Portal Server Access Manager
Directory Server	Keine	Portal Server Calendar Server Messaging Server Instant Messaging Access Manager

## Dimension 2: Logische Schichten

Die interagierenden Softwarekomponenten verteilter Unternehmensanwendungen können logischen Schichten zugeordnet werden. Diese Schichten stellen die logische und physische Unabhängigkeit von Softwarekomponenten auf der Grundlage der Art der von ihnen angebotenen Dienste dar.

Die Dimension der logischen Schichten in einer Architektur ist in der folgenden Abbildung dargestellt.

**Abbildung 2-4** Dimension 2: Logische Schichten für verteilte Unternehmensanwendungen

Logische Schichtenarchitekturen stellen überwiegend die in [Abbildung 1-1 auf Seite 19](#) dargestellte Ebene der verteilten Unternehmensanwendung dar.

Die unter „[Infrastrukturdienstebenen](#)“ auf [Seite 37](#) behandelten Java ES-Systemdienstkomponenten unterstützen Anwendungskomponenten aller in [Abbildung 2-4](#) dargestellten logischen Schichten. Das Konzept der logischen Schichten gilt jedoch ebenso für Systemdienstkomponenten, die Dienste auf Anwendungsebene bereitstellen, wie Messaging Server und Calendar Server.

## Beschreibung der logischen Schichten

Dieser Abschnitt enthält Kurzbeschreibungen der vier logischen Schichten, die in [Abbildung 2-4](#) dargestellt sind. Die Beschreibungen beziehen sich auf Anwendungskomponenten, die mit einem auf der Java 2 Platform Enterprise Edition (J2EE™-Plattform) basierenden Komponentenmodell implementiert wurden. Andere verteilte Komponentenmodelle hingegen, beispielsweise CORBA, unterstützen ebenfalls diese Architektur.

- **Client-Schicht:** Die Client-Schicht umfasst Anwendungslogik, auf die ein Endbenutzer über eine Benutzeroberfläche direkt zugreift. Die Logik in der Client-Schicht kann browserbasierte Clients, auf einem Desktop-Computer ausgeführte Java-Komponenten oder auf einem Handheld-Gerät ausgeführte mobile Java 2 Platform Micro Edition-Clients (J2ME™-Plattform) enthalten.
- **Präsentationsschicht:** Die Präsentationsschicht umfasst Anwendungslogik, die Daten für die Zustellung an die Client-Schicht vorbereitet und Anforderungen der Client-Schicht für die Zustellung an die Back-End-Geschäftslogik verarbeitet. Die Logik in der Präsentationsschicht umfasst in der Regel J2EE-Komponenten, wie Java Servlet-Komponenten oder JSP-Komponenten, die Daten für die Zustellung im HTML- oder XML-Format vorbereiten oder Verarbeitungsanforderungen empfangen. Diese Schicht kann auch einen Zugangsdienst enthalten, der einen personalisierten, sicheren und benutzerdefinierten Zugriff auf die in der Geschäftsdienstschicht vorhandenen [Geschäftsdienste](#) ermöglichen kann.

- **Geschäftsdienstschiicht:** Die Geschäftsdienstschiicht umfasst die Logik, die die Hauptfunktionen der Anwendung ausführt: Datenverarbeitung, Implementierung der Geschäftsregeln, Koordination mehrerer Benutzer und Verwaltung externer Ressourcen, wie Datenbanken oder Legacy-Systeme. In der Regel umfasst diese Schicht eng miteinander verknüpfte Komponenten, die dem verteilten J2EE-Komponentenmodell entsprechen, wie Java-Objekte, EJB-Komponenten oder nachrichtengesteuerte Beans (MDBs). Einzelne J2EE-Komponenten können für die Bereitstellung komplexer Geschäftsdienste, beispielsweise Inventardienste oder Steuerberechnungsdienste, zusammengefügt werden. Einzelne Komponenten und Dienstgruppen können in einem dienstorientierten Architekturmodell zu lose miteinander verknüpften *Webdiensten* zusammengefasst werden, die dem Schnittstellenstandard SOAP (Simple Object Access Protocol) entsprechen. Geschäftsdienste können auch als eigenständige *Server*, beispielsweise als Unternehmenskalenderserver oder Messaging-Server, erstellt werden.
- **Datenschicht:** Die Datenschicht besteht aus Diensten, die von der Geschäftslogik genutzte persistente Daten liefern. Die Daten können Anwendungsdaten sein, die in einem Datenbankverwaltungssystem gespeichert sind, oder Ressourcen- und Verzeichnisinformationen, die in einem Lightweight Directory Access Protocol-Datenspeicher (LDAP-Datenspeicher) gespeichert sind. Die Datendienste können auch Daten aus externen Quellen oder Daten aus Legacy-Computersystemen enthalten.

## Logische und physische Unabhängigkeit

Die in [Abbildung 2-4 auf Seite 43](#) illustrierte Architekturdimension betrachtet vor allem die logische und physische Unabhängigkeit von Komponenten, die durch vier separate Schichten dargestellt wird. Diese Schichten stellen die Partitionierung der Anwendungslogik über die verschiedenen in einer Netzwerkumgebung vorhandenen Computer hinweg dar:

- **Logische Unabhängigkeit:** Die vier Schichten des Architekturmodells stellen die logische Unabhängigkeit dar: Sie können die Anwendungslogik in einer Schicht (beispielsweise in der Geschäftsdienstschiicht) unabhängig von der Logik in den anderen Schichten ändern. Genauso können Sie Ihre Implementierung der Geschäftslogik ändern, ohne die Logik in der Präsentations- oder Client-Schicht ändern oder aktualisieren zu müssen. Diese Unabhängigkeit bedeutet, dass Sie beispielsweise neue Typen von Client-Komponenten aufnehmen können, ohne die Geschäftsdienstkomponenten zu ändern.

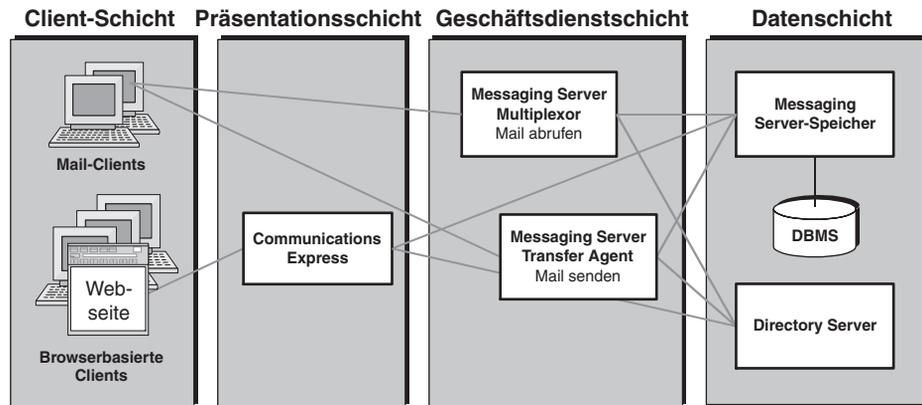
- **Physische Unabhängigkeit:** Die vier Schichten stellen auch die physische Unabhängigkeit dar: Sie können die Logik in vier verschiedenen Schichten auf verschiedenen Hardwareplattformen (verschiedene CPU-Konfigurationen, Chipsätze und Betriebssysteme) bereitstellen. Diese Unabhängigkeit ermöglicht es, verteilte Anwendungskomponenten auf den Computern auszuführen, die den einzelnen Computeranforderungen am ehesten entsprechen und am besten zur Maximierung der Netzwerkbandbreite geeignet sind.

Wie Sie Anwendungs- oder Infrastrukturkomponenten auf eine Hardwareumgebung (Ihre Bereitstellungsarchitektur) abbilden, hängt von vielen Faktoren ab, die durch die Größe und Komplexität Ihrer Softwarelösung bestimmt werden. Bei sehr kleinen Bereitstellungen kann eine Bereitstellungsarchitektur nur wenige Computer betreffen. Bei großen Bereitstellungen kann die Zuordnung von Komponenten zu einer Hardwareumgebung Faktoren wie Geschwindigkeit und Leistung einzelner Computer, Geschwindigkeit und Bandbreite von Netzwerkleitungen, Sicherheits- und Firewall-Aspekte sowie Replikationsstrategien für Komponenten berücksichtigen.

## Auf Systemkomponenten angewendete geschichtete Architektur

Wie in [Abbildung 2-3 auf Seite 41](#) dargestellt, bieten Java ES-Infrastrukturdienstkomponenten die zugrundeliegende Infrastrukturunterstützung für verteilte Softwarelösungen. Einige dieser Lösungen enthalten jedoch Dienste der Anwendungsebene, die direkt von Java ES-Komponenten realisiert werden. Diese Lösungen verwenden den konzeptionellen Ansatz der logischen Schichten.

Beispiel: Die von Messaging Server bereitgestellten E-Mail-Kommunikationsdienste sind so implementiert, dass sie bestimmte logisch gesonderte Konfigurationen von Messaging Server verwenden. Diese gesonderten Konfigurationen bieten jeweils einen gesonderten Satz von Diensten. Bei der Konzeption einer Messaging-Lösung werden diese gesonderten Konfigurationen, wie die folgende Abbildung zeigt, als separate Komponenten dargestellt, die sich auf unterschiedlichen logischen Schichten befinden.

**Abbildung 2-5** Messaging Server: Beispiel für eine Schichtenarchitektur

**HINWEIS** [Abbildung 2-3](#) darf nicht als vollständige logische Architektur verstanden werden. Einige Java ES-Komponenten wurden weggelassen, um die Abbildung zu vereinfachen. Die zwischen den Komponenten verlaufenden Linien stellen Interaktionen dar.

Die logische Trennung von Messaging Server-Funktionen in verschiedene Schichten erlaubt logisch unterschiedliche Konfigurationen von Messaging Server, die auf verschiedenen Computern in einer physischen Umgebung bereitgestellt werden. Die physische Trennung gestattet eine größere Flexibilität bei der Einhaltung der Dienstqualitätsanforderungen (siehe „[Dimension 3: Dienstqualität](#)“). Dies bietet beispielsweise unterschiedliche Verfügbarkeitslösungen für unterschiedliche Instanzen und unterschiedliche Sicherheitsimplementierungen für unterschiedliche Messaging Server-Funktionen.

## Dimension 3: Dienstqualität

Die beiden vorherigen Architekturdimensionen (Infrastrukturdienstabhängigkeiten und logische Schichten) betreffen überwiegend die logischen Aspekte der Architektur, nämlich, welche Komponenten auf welche Weise für die Interaktion erforderlich sind, um Dienste für Endbenutzer bereitzustellen. Eine ebenso wichtige Dimension jeder bereitgestellten Lösung ist jedoch die Fähigkeit, Dienstqualitätsanforderungen zu erfüllen.

Die Dimension der Dienstqualität einer Lösungsarchitektur stellt die Rolle der Java ES-Dienstqualitätskomponenten in den Vordergrund.

## Dienstqualitäten

Aufgrund der Tatsache, dass Internet- und E-Commerce-Dienste für Geschäftsvorgänge immer wichtiger werden, stellt die Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit dieser Dienste eine der wichtigsten Dienstqualitätsanforderungen umfangreicher, extrem leistungsstarker Bereitstellungsarchitekturen dar.

Für die Konzeption einer erfolgreichen Softwarelösung müssen Sie relevante Dienstqualitätsanforderungen aufstellen und eine Architektur entwerfen, die diese erfüllt. Um Dienstqualitätsanforderungen zu definieren, werden einige wichtige Dienstqualitäten verwendet. Die folgende Tabelle führt diese Dienstqualitäten auf.

**Tabelle 2-2** Dienstqualitäten mit Auswirkung auf die Lösungsarchitektur

Systemdienstqualitäten	Beschreibung
Leistung	Die Messung der Antwortzeit und -latenz in Bezug auf die Benutzerladebedingungen.
Verfügbarkeit	Ein Maß dafür, wie oft die Ressourcen und Dienste eines Systems für Endbenutzer verfügbar sind (die <i>Betriebszeit</i> eines Systems).
Sicherheit	Eine komplexe Kombination von Faktoren, die die Integrität eines Systems und seiner Benutzer beschreibt. Zur Sicherheit gehören die physische Sicherheit der Systeme, die Netzwerksicherheit, die Anwendungs- und Datensicherheit (Authentifizierung und Autorisierung der Benutzer) sowie der sichere Transport von Informationen.
Skalierbarkeit	Die Möglichkeit, einem bereitgestellten System im Laufe der Zeit Kapazität hinzuzufügen. Die Skalierbarkeit umfasst in der Regel das Hinzufügen von Ressourcen zum System, sollte jedoch keine Änderungen an der Bereitstellungsarchitektur erfordern.
Latente Kapazität	Die Fähigkeit eines Systems, eine außergewöhnliche Spitzenauslastung ohne zusätzliche Ressourcen zu bewältigen.
Zweckmäßigkeit	Die Einfachheit der Wartung eines bereitgestellten Systems, einschließlich der Überwachung des Systems, dem Beheben von auftretenden Problemen und der Aufrüstung der Hardware- und Softwarekomponenten.

Die Dimension der Dienstqualität wirkt sich stark auf die Bereitstellungsarchitektur einer Lösung aus: Wie Anwendungskomponenten und Infrastrukturkomponenten in einer physischen Umgebung bereitgestellt werden.

Die Dienstqualitäten, die die Bereitstellungsarchitektur beeinflussen, sind eng miteinander verbunden: Anforderungen an eine Systemqualität wirken sich häufig auf die Konzeption der anderen Dienstqualitäten aus. So kann beispielsweise ein höheres Sicherheitsniveau die Leistung beeinträchtigen, was wiederum die Verfügbarkeit beeinflusst. Das Hinzufügen weiterer Computer, um Verfügbarkeitsprobleme durch Redundanz zu beheben, wirkt sich häufig auf die Wartungskosten (Zweckmäßigkeit) aus.

Das Verständnis der Beziehungen zwischen den Dienstqualitäten und ihre Abstimmung ist eine wichtige Voraussetzung für die Konzeption von Architekturen, die sowohl die Geschäftsanforderungen als auch die Geschäftsbeschränkungen einhalten.

## Java Enterprise System-Dienstqualitätskomponenten

Einige Java ES-Komponenten werden hauptsächlich verwendet, um die Dienstqualitäten von Systemdienstkomponenten oder verteilten Anwendungskomponenten zu verbessern. Diese Softwarekomponenten werden oft zusammen mit Hardwarekomponenten, wie Lastausgleichsmodulen und Firewalls, eingesetzt.

Die unter „[Dienstqualitätskomponenten](#)“ auf [Seite 23](#) vorgestellten Java ES-Dienstqualitätskomponenten können wie folgt zusammengefasst werden:

- **Verfügbarkeitskomponenten:** Diese Komponenten sorgen für eine nahezu kontinuierliche Betriebszeit der bereitgestellten Lösung.
- **Zugriffskomponenten:** Diese Komponenten sorgen für einen sicheren Internet-Zugriff auf Systemdienste und bieten häufig auch eine Routing-Funktion.
- **Verwaltungskomponenten:** Diese Komponenten sorgen für eine Verbesserung der Zweckmäßigkeit von Systemkomponenten.

Die folgende Tabelle enthält die wichtigsten Java ES-Dienstqualitätskomponenten aus der Perspektive der Architektur und zeigt, auf welche Systemqualitäten sie sich am meisten auswirken.

**Tabelle 2-3** Dienstqualitätskomponenten und die beeinflussten Systemqualitäten

Komponente	Beeinflusste Systemqualitäten
Communications Express	Sicherheit Skalierbarkeit
Directory Proxy Server	Sicherheit Skalierbarkeit

**Tabelle 2-3** Dienstqualitätskomponenten und die beeinflussten Systemqualitäten

<b>Komponente</b>	<b>Beeinflusste Systemqualitäten</b>
High Availability Session Store	Verfügbarkeit
Portal Server Secure Remote Access	Sicherheit Skalierbarkeit
Sun Cluster	Verfügbarkeit Skalierbarkeit
Sun Remote Services Net Connect	Zweckmäßigkeit

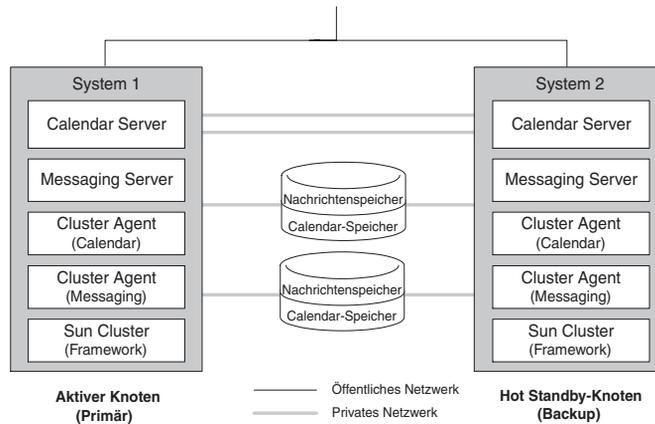
## Sun Cluster-Software

Die Sun Cluster-Software bietet Hochverfügbarkeits- und Skalierbarkeitsdienste für Java ES sowie für Anwendungen, die von der Java ES-Infrastruktur unterstützt werden.

Ein Cluster besteht aus einer Reihe von lose miteinander verknüpfter Computern, die gemeinsam eine einzige Client-Ansicht für Dienste, Systemressourcen und Daten bereitstellen. Intern verwendet der Cluster redundante Computer, Interconnects, Datenspeicher und Netzwerkschnittstellen zur Bereitstellung der Hochverfügbarkeit für clusterbasierte Dienste und Daten.

Die Sun Cluster-Software überwacht permanent den Zustand der Mitglieds-knoten und anderer Cluster-Ressourcen. Im Fehlerfall greift die Sun Cluster-Software ein und löst das Failover der überwachten Ressourcen aus, wobei die interne Redundanz genutzt wird, um einen nahezu kontinuierlichen Zugriff auf diese Ressourcen zu realisieren.

Die folgende Abbildung zeigt einen Zwei-Knoten Cluster, der Datenspeicherdienste für Messaging Server und Calendar Server unterstützt.

**Abbildung 2-6** Verfügbarkeitskonzept mit Sun Cluster-Knoten

Sun Cluster-Datendienstpakete (auch als Sun Cluster-Agenten bezeichnet) sind für alle Java ES-Systemdienstkomponenten verfügbar. Sie können auch für selbst entwickelte Anwendungskomponenten Agenten schreiben.

Aufgrund der Steuerung durch die Sun Cluster-Software kann ein Cluster auch skalierbare Dienste bereitstellen. Durch Nutzung des globalen Dateisystems des Clusters und wegen der Fähigkeit, Infrastruktur- oder Anwendungsdienste auf mehreren Knoten in einem Cluster auszuführen, kann eine verstärkte Anforderung dieser Dienste auf mehrere Instanzen der Dienste aufgeteilt werden. Daher kann Sun Cluster-Software bei richtiger Konfigurierung in einer verteilten Unternehmensanwendung gleichzeitig Hochverfügbarkeit und Skalierbarkeit gewährleisten.

Aufgrund der für eine Sun Cluster-Umgebung erforderlichen Redundanz sorgt die Aufnahme von Sun Cluster in einer Lösung für eine merkliche Erhöhung der in der physischen Umgebung benötigten Anzahl von Computern und Netzwerkverbindungen.

Anders als die von anderen Java ES-Komponenten bereitgestellten Dienste, handelt es sich bei den Verfügbarkeitsdiensten von Sun Cluster um verteilte Peer-to-Peer-Dienste. Deshalb muss die Sun Cluster-Software auf jedem, in einem Cluster vorhandenen Computer installiert werden.

## Synthese der drei Architekturdimensionen

Zusammen genommen bilden die drei in [Abbildung 2-1](#) dargestellten und in den letzten Abschnitten besprochenen Architekturdimensionen ein Framework für die Konzeption verteilter Softwarelösungen. Die drei Dimensionen (Infrastrukturdienstabhängigkeiten, logische Schichten und Dienstqualität) verdeutlichen die von den Java ES-Komponenten in Lösungsarchitekturen gespielte Rolle.

Jede Dimension steht für eine unterschiedliche Architekturperspektive. Eine Lösungsarchitektur muss alle Dimensionen berücksichtigen. Beispiel: Verteilte Komponenten in jeder logischen Schicht einer Lösungsarchitektur (Dimension 2) müssen durch entsprechende Infrastrukturkomponenten (Dimension 1) und entsprechende Dienstqualitätskomponenten (Dimension 3) unterstützt werden.

Entsprechend spielt jede in einer Lösungsarchitektur vorhandene Komponente hinsichtlich der verschiedenen Architekturdimensionen unterschiedliche Rollen. Directory Server kann beispielsweise sowohl als Back-End-Komponente der Datenschicht (Dimension 2) als auch als Anbieter von Persistenzdiensten (Dimension 1) angesehen werden.

Wegen der Zentralität von Directory Server hinsichtlich dieser beiden Dimensionen stehen Dienstqualitätsprobleme (Dimension 3) für diese Java ES-Komponente an höchster Stelle. Ein Directory Server-Ausfall würde schwerwiegende Auswirkungen auf ein Geschäftssystem haben, daher ist die Konzeption der Hochverfügbarkeit für diese Komponente sehr wichtig. Und da Directory Server zum Speichern sensibler Benutzer- oder Konfigurationsinformationen verwendet wird, ist für diese Komponente auch das Sicherheitskonzept sehr wichtig.

Das Zusammenspiel der drei Dimensionen hinsichtlich der Java ES-Komponenten wirkt sich auf die Konzeption der logischen Architektur und der Bereitstellungsarchitektur der Lösungen aus.

Die detaillierte Beschreibung der Entwurfsmethodik für die Verwendung des in [Abbildung 2-1](#) dargestellten Architekturframeworks würde den Rahmen dieses Handbuchs sprengen. Das dreidimensionale Architekturframework hebt jedoch Aspekte der Konzeption hervor, die für das Verständnis des Bereitstellens von auf Java Enterprise System basierenden Softwarelösungen wichtig sind.

# Beispiel für Java Enterprise System-Lösungsarchitektur

Java Enterprise System unterstützt eine Vielzahl von Softwarelösungen.

Viele Lösungen können mit den in Java Enterprise System enthaltenen Komponenten ohne Entwicklungsaufwand direkt konzipiert und bereitgestellt werden. Bei anderen Lösungen können umfangreiche Entwicklungsarbeiten notwendig sein, bei denen Sie eigene J2EE-Komponenten entwickeln müssen, die neue Geschäfts- oder Präsentationsdienste bereitstellen. Sie können die selbst entwickelten Komponenten als Webdienste zusammenfassen, die dem Schnittstellenstandard SOAP (Simple Object Access Protocol) entsprechen. Bei vielen Lösungen ist eine Kombination dieser beiden Ansätze erforderlich.

In diesem Abschnitt finden Sie ein aus den im letzten Abschnitt beschriebenen Architekturkonzepten abgeleitetes Beispiel, das verdeutlicht, wie Java Enterprise System eine Out-of-the-Box-Lösung unterstützt.

## Szenario der Unternehmenskommunikation

Unternehmen müssen in der Regel die Kommunikation zwischen Ihren Mitarbeitern, insbesondere durch E-Mail- und Kalenderdienste sicherstellen. Solche Unternehmen wollen, dass ihre Mitarbeiter einen personalisierten Zugang zu internen Websites und anderen Ressourcen besitzen, der auf den unternehmensweiten Authentifizierungs- und Autorisierungsdiensten basiert. Zusätzlich wollen diese Unternehmen, dass die Identität der Mitarbeiter über alle Unternehmensdienste hinweg protokolliert wird, damit eine einzige Webanmeldung (Single Sign-On) den Zugriff auf alle Dienste bietet.

Diese spezifischen Geschäftsanforderungen, die lediglich eine Beispielmenge der gesamten Geschäftsanforderungen darstellen, sind in der folgenden Tabelle zusammengefasst.

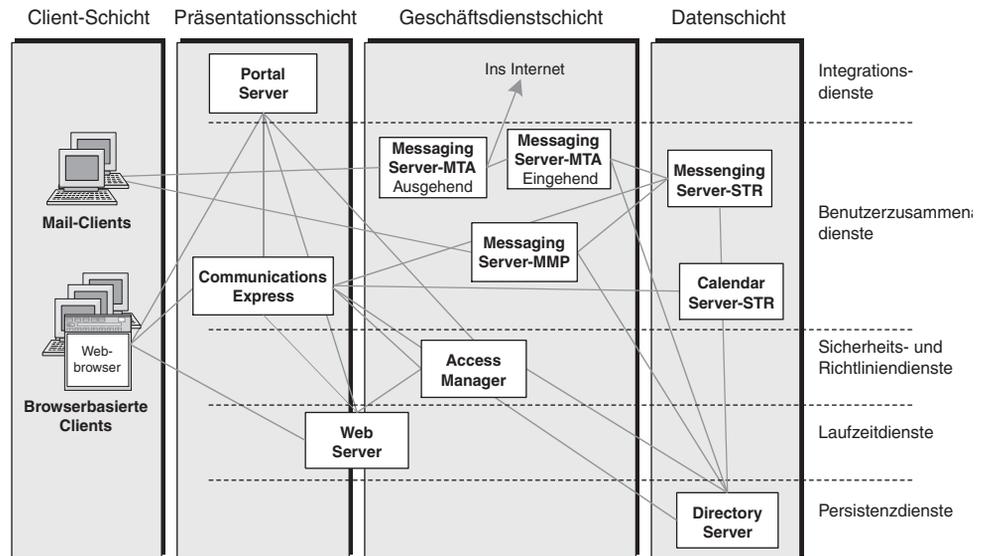
**Tabelle 2-4** Zusammenfassung der Geschäftsanforderungen: Kommunikationsszenario

<b>Unternehmensanforderung</b>	<b>Beschreibung</b>	<b>Benötigte Java ES-Dienste</b>
Single Sign-On	Zugang zu sicheren Unternehmensressourcen und -diensten auf der Grundlage einer einzigen Identität mit Single Sign-On für Webzugang.	Identitätsdienste
Messaging Kalender	E-Mail-Messaging zwischen Mitarbeitern und mit der Außenwelt. Elektronische Kalendereinträge und Terminvereinbarungen für Mitarbeiter.	Kommunikations- und Zusammenarbeitsdienste
Portal-Zugriff	Einzelner, webbasierter, personalisierter Zugangspunkt für Kommunikationsdienste, wie E-Mail und Kalender sowie interne Webseiten.	Zugangsdienste

Außerdem hat ein Unternehmen darüber hinaus Anforderungen hinsichtlich der Leistung, Verfügbarkeit, Netzwerksicherheit und Skalierbarkeit des Softwaresystems, das diese Dienste bereitstellt.

## Logische Architektur des Beispielszenarios

Eine logische Architektur für die Bereitstellung der in [Tabelle 2-4](#) aufgeführten Zugangs-, Kommunikations- und Identitätsdienste mit Java ES-Komponenten enthält die folgende Abbildung. Die Architektur behandelt logisch getrennte Konfigurationen von Messaging Server als separate Komponenten, da diese unterschiedliche Dienste bereitstellen.

**Abbildung 2-7** Logische Architektur des Unternehmenskommunikationsszenarios

Die Komponenten sind in einer horizontalen Dimension platziert, die die standardmäßigen logischen Schichten darstellen. Die vertikale Dimension stellt die Infrastrukturdienstebenen dar. Die Interaktionen zwischen den Komponenten hängen entweder von ihrer Funktion als verteilte Infrastrukturdienste ab (Interaktionen zwischen Infrastrukturdienstebenen) oder von ihrer Rolle in einer Schichtenarchitektur der Anwendung (Interaktionen innerhalb und zwischen logischen Schichten).

In dieser Architektur arbeitet Access Manager, der auf die in Directory Server gespeicherten Benutzerinformationen zugreift, als Schiedsrichter, wenn es um die Authentifizierung und Autorisierung über Single Sign-On für den Portal Server und andere webbasierte Komponenten der Präsentationsschicht geht. Zu den Messaging Server-Komponenten gehört in der Datenschicht ein Nachrichtenspeicher (Messaging Server-STR), der Komponenten der Geschäftsdienstschicht sendet und abrufen, sowie in der Präsentationsschicht eine HTTP-Zugangskomponente und Communications Express.

Die logische Architektur zeigt außerdem die Infrastrukturdienstabhängigkeiten zwischen den verschiedenen Java ES-Komponenten. So hängt beispielsweise Portal Server für die Messaging- und Kalenderkanäle von Communications Express ab und für die Authentifizierungs- und Autorisierungsdienste von Access Manager. Diese Komponenten hängen wiederum von Directory Server ab, von dem Sie Benutzerinformationen und Konfigurationsdaten erhalten. Einige Komponenten benötigen Webcontainerdienste, die Web Server bereitstellt.

Weitere Informationen über das logische Konzept der Java ES-Lösung finden Sie im *Java Enterprise System Deployment Planning Guide*.

## Bereitstellungsarchitektur des Beispielszenarios

Beim Übergang von der logischen Architektur zur Bereitstellungsarchitektur sind die Dienstqualitätsanforderungen von größter Bedeutung. So können beispielsweise geschützte Subnetze und Firewalls genutzt werden, um eine Sicherheitsbarriere zum Back-End-Bereich zu erzeugen. Für viele Komponenten müssen Verfügbarkeits- und Skalierbarkeitsanforderungen erfüllt werden, indem diese auf mehreren Computern bereitgestellt und die Anfragen über ein Lastausgleichsmodul an die replizierten Komponenten verteilt werden.

Wenn jedoch strengere Verfügbarkeitsanforderungen vorliegen und wenn sehr viel Festplattenspeicher benutzt wird, dann sind andere Verfügbarkeitslösungen besser geeignet. Für den Messaging Server-Speicher kann beispielsweise Sun Cluster und für Directory Server die Multi-Master-Replikation genutzt werden.

Weitere Informationen über das Bereitstellungs-konzept der Java ES-Lösung finden Sie im *Java Enterprise System Deployment Planning Guide*.

# In diesem Kapitel enthaltene Schlüsselbegriffe

Dieser Abschnitt erläutert in diesem Kapitel verwendete wichtige technische Begriffe, wobei der Schwerpunkt darauf liegt, die Beziehungen zwischen diesen Begriffen und ihrer Verwendung im Java Enterprise System-Kontext zu verdeutlichen.

**Anwendungskomponente** Eine kundenspezifisch entwickelte Software-*Komponente*, die einige spezifische Datenverarbeitungsfunktionen durchführt und *Endbenutzer* oder andere Anwendungskomponenten mit *Geschäftsdiensten* versorgt. Eine Anwendungskomponente entspricht in der Regel einem Modell verteilter Komponenten (wie CORBA und die J2EE™-Plattform). Diese Komponenten können einzeln oder kombiniert zu *Webdiensten* zusammengefasst werden.

**Architektur** Ein Konzept, das die logischen und physischen modularen Blöcke einer verteilten Anwendung (oder eines anderen Softwaresystems) sowie ihre Beziehungen untereinander darstellt. Für eine *Verteilte Unternehmensanwendung* umfasst das Architekturkonzept im Allgemeinen die *logische Architektur* und die *Bereitstellungsarchitektur* der Anwendung.

**Geschäftsdienst** Eine *Anwendungskomponente* oder eine Komponentengruppe, die die Geschäftslogik im Namen mehrerer Clients ausführt (und daher einen Vorgang mit mehreren Threads darstellt). Ein Geschäftsdienst kann auch eine Gruppe von verteilten Komponenten, die zu einem *Webdienst* zusammengefasst sind, oder ein eigenständiger *Server* sein.

**Client** Eine Software, die Software-*Dienste* anfordert. (Hinweis: Hierbei handelt es sich nicht um eine Person – siehe *Endbenutzer*.) Ein Client kann ein Dienst sein, der einen anderen Dienst anfordert, oder eine GUI-Komponente, auf die ein Endbenutzer zugreift.

**Bereitstellungsarchitektur** Ein starkes Konzept, das die *logische Architektur* einer physischen Computerumgebung zuordnet. Die physische Umgebung umfasst die in einer Intranet- oder Internetumgebung vorhandenen Computer, die Netzwerkverbindungen zwischen ihnen sowie andere physische Geräte, die zur Unterstützung der Software erforderlich sind.

**logische Architektur** Ein Konzept, das die logischen modularen Blöcke einer verteilten Anwendung sowie ihre Beziehungen untereinander (bzw. ihre Schnittstellen) darstellt. Die logische Architektur umfasst sowohl die verteilten *Anwendungskomponenten* als auch die Infrastrukturdienstkomponenten, die für deren Unterstützung erforderlich sind.

**Server** Ein Softwarevorgang mit mehreren Threads (im Gegensatz zu einem Hardwareserver), der einen verteilten *Dienst* oder eine geschlossene Gruppe von Diensten für *Clients* bereitstellt, die über eine externe Schnittstelle auf den Dienst zugreifen.

**Webdienst** Ein Dienst, der den standardisierten Internetprotokollen für Verfügbarkeit, Dienstintegration und Erkennung entspricht. Zu diesen Standards gehören das SOAP-Nachrichtenprotokoll (Simple Object Access Protocol), die WSDL-Schnittstellendefinition (Web Service Definition Language) und der UDDI-Registrierungsstandard (Universal Discovery, Description and Integration).

# Integrationsfunktionen von Java Enterprise System

Dieses Kapitel beschreibt den konzeptionellen und technischen Hintergrund der Funktionen, die bei der Integration von Java ES-Komponenten zu einem einzigen Softwaresystem eine Schlüsselrolle spielen.

Anhand dieser Funktionen werden die Vorteile von Java Enterprise System im Vergleich zur manuellen Integration unvereinbarer Infrastrukturprodukte deutlich.

Das Kapitel behandelt folgende Funktionen:

- „Das integrierte Installationsprogramm von Java Enterprise System“ auf Seite 60
- „Integrierte Identitäts- und Sicherheitsdienste“ auf Seite 62
- „In diesem Kapitel enthaltene Schlüsselbegriffe“ auf Seite 68

# Das integrierte Installationsprogramm von Java Enterprise System

Alle Java ES-Komponenten werden mithilfe eines einzigen Installationsprogramms installiert. Dieses Installationsprogramm bietet für alle Komponenten konsistente Verfahren zur Installation und Deinstallation und weist ein konsistentes Verhalten auf.

Beim Java ES-Installationsprogramm handelt es sich um ein integriertes Framework zur Übertragung von Java ES-Software auf ein Hostsystem. Mit dem Installationsprogramm können Sie bestimmte Java ES-Komponenten auswählen und auf beliebigen Computern Ihrer Datenverarbeitungsumgebung installieren. Das Installationsprogramm bietet auch einige Möglichkeiten zur Konfiguration des Installationszeitpunkts, die von den einzelnen zu installierenden Java ES-Komponenten abhängen.

Das Java ES-Installationsprogramm führt selbstständig keine verteilten Installationen durch. Um eine verteilte Java ES-Softwarelösung bereitzustellen, installieren Sie die entsprechenden Komponenten mit dem Java ES-Installationsprogramm nacheinander auf jedem Computer Ihrer Umgebung. Abhängig von Ihrer Bereitstellungsarchitektur und den Abhängigkeiten zwischen den Komponenten müssen Sie eine Reihe von Installationssitzungen und Konfigurationsschritten durchführen.

Das Installationsprogramm arbeitet interaktiv sowohl in einem Grafikmodus als auch in einem textbasierten Modus und verfügt außerdem über einen parametergesteuerten Modus zur automatischen Installation. Neben der englischen Version unterstützt das Installationsprogramm sieben weitere Sprachen: Französisch, Deutsch, Spanisch, Koreanisch, Chinesisch (vereinfacht), Chinesisch (traditionell) und Japanisch.

In diesem Abschnitt werden folgende Aspekte des integrierten Installationsprogramms für Java ES behandelt (genauere Informationen finden Sie im *Java Enterprise System Installation Guide*):

- [Überprüfung bereits vorhandener Software](#)
- [Überprüfung der Abhängigkeiten](#)
- [Erstkonfiguration](#)
- [Deinstallation](#)

## Überprüfung bereits vorhandener Software

Das Installationsprogramm überprüft den Computer, auf dem die Installation durchgeführt wird, und ermittelt die bereits installierten Komponenten von Java ES. Das Installationsprogramm führt dann Prüfungen auf verschiedenen Ebenen durch, um sicherzustellen, dass alle vorhandenen Komponenten die für eine erfolgreiche Zusammenarbeit erforderliche Versionsstufe aufweisen. Das Installationsprogramm informiert Sie darüber, welche Softwarekomponenten nicht kompatibel sind und aufgerüstet oder entfernt werden müssen.

Ebenso überprüft das Installationsprogramm, ob bereits gemeinsam genutzte Java ES-Komponenten installiert wurden (siehe „[Gemeinsam genutzte Komponenten](#)“ auf Seite 26), wie beispielsweise J2SE oder NSS. Wenn das Installationsprogramm gemeinsam genutzte Komponenten in nicht kompatiblen Versionen findet, werden diese aufgelistet. Wenn Sie mit der Installation fortfahren, rüstet das Installationsprogramm die gemeinsam genutzten Komponenten auf eine neuere Version auf.

## Überprüfung der Abhängigkeiten

Das Installationsprogramm führt eine intensive Überprüfung der Komponenten durch, um sicherzustellen, dass die von Ihnen ausgewählten Installationskomponenten ordnungsgemäß funktionieren.

Etliche Komponenten sind von anderen Komponenten abhängig. Das Installationsprogramm stellt die Logik zur Verfügung, die zur Erfüllung dieser Abhängigkeitsanforderungen erforderlich ist. Aus diesem Grund schließt das Installationsprogramm, wenn Sie eine Komponente für die Installation auswählen, automatisch die Komponenten und Unterkomponenten mit ein, zu denen die ausgewählte Komponente in einem Abhängigkeitsverhältnis steht.

Sie können die Auswahl einer Komponente nicht aufheben, wenn eine andere ausgewählte Komponente lokal von dieser Komponente abhängt. Wenn die Abhängigkeit jedoch nicht auf lokaler Ebene besteht, wird eine Warnmeldung ausgegeben. Sie können den Vorgang jedoch fortsetzen (es wird davon ausgegangen, dass die Abhängigkeitsanforderung durch eine Komponente auf einem anderen Hostcomputer erfüllt wird).

## Erstkonfiguration

Bei vielen Java ES-Komponenten ist eine Erstkonfiguration erforderlich, damit diese gestartet werden können. Bei einigen Komponenten kann das Java ES-Installationsprogramm diese Erstkonfiguration vornehmen.

Sie können auswählen, ob das Installationsprogramm die Erstkonfiguration durchführen (Option „Jetzt konfigurieren“) oder die Software ohne Erstkonfiguration installieren soll (Option „Später konfigurieren“). Im letzten Fall müssen Sie jede Komponente nach Abschluss der Installation explizit konfigurieren.

Wenn Sie auswählen, dass das Installationsprogramm die Erstkonfiguration vornehmen soll, müssen Sie während des Installationsvorgangs die benötigten Konfigurationsangaben vornehmen. Insbesondere können Sie eine Reihe von Parameterwerten angeben, die für alle Produktkomponenten gelten sollen, beispielsweise eine Administrator-ID und das zugehörige Passwort.

## Deinstallation

Zu Java Enterprise System gehört außerdem ein Deinstallationsprogramm. Mit diesem Programm können Sie die Komponenten entfernen, die vom Java ES-Installationsprogramm auf dem lokalen Computer installiert wurden. Das Deinstallationsprogramm prüft, ob lokale Abhängigkeiten vorliegen, und gibt gegebenenfalls eine Warnmeldung aus. Das Deinstallationsprogramm entfernt keine gemeinsam genutzten Java ES-Komponenten.

Das Deinstallationsprogramm kann, ebenso wie das Installationsprogramm, im grafischen, textbasierten oder Automatikmodus ausgeführt werden.

# Integrierte Identitäts- und Sicherheitsdienste

Eine wichtige Funktion von Java Enterprise System besteht in der integrierten Verwaltung von Benutzeridentitäten und dem zugehörigen integrierten Authentifizierungs- und Autorisierungs-Framework.

In den folgenden Abschnitten finden Sie den technischen Hintergrund für das Verständnis der integrierten Identitäts- und Sicherheitsdienste, die Java Enterprise System bietet:

- [Einzelidentität](#)
- [Authentifizierung und Single Sign-On](#)

## Einzelidentität

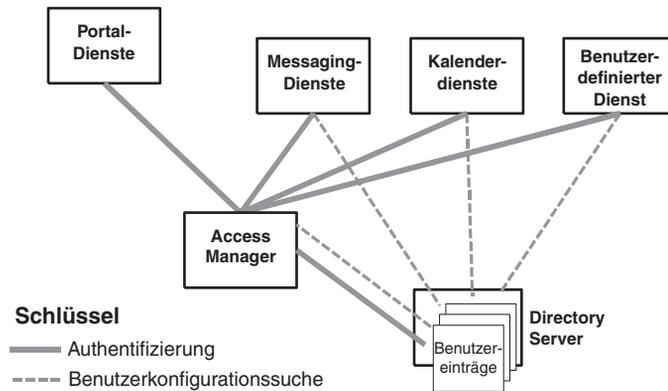
Innerhalb einer Java ES-Umgebung verfügt jeder Benutzer über eine einzelne integrierte Identität. Anhand dieser *Einzelidentität* kann dem Benutzer der Zugriff auf verschiedene Ressourcen gestattet werden, beispielsweise Portale, Webseiten und Dienste wie Nachrichtendienste, Kalenderdienste und Instant Messaging.

Diese integrierte Identitäts- und Sicherheitsfunktion beruht auf einer engen Zusammenarbeit zwischen Directory Server, Access Manager und anderen Komponenten von Java ES.

Der Benutzerzugriff auf einen Dienst bzw. eine Ressource von Java ES wird gewährt, indem die benutzerspezifischen Informationen in einem einzelnen Benutzereintrag in einem Benutzer-Repository oder *Verzeichnis* gespeichert werden. Zu diesen Informationen gehören normalerweise Angaben wie ein eindeutiger Name und ein Passwort, eine E-Mail-Adresse, die Funktion innerhalb einer Organisation, Webseiteneinstellungen usw. Die Informationen im Benutzereintrag können zur Authentifizierung des Benutzers, zur Autorisierung des Zugriffs auf bestimmte Ressourcen und zur Bereitstellung verschiedener Dienste für den jeweiligen Benutzer verwendet werden.

Bei Java Enterprise System werden die Benutzereinträge in einem von Directory Server bereitgestellten Verzeichnis gespeichert. Wenn ein Benutzer einen von einer Java ES-Komponente bereitgestellten Dienst anfordern möchte, verwendet dieser Dienst Access Manager für die Authentifizierung des Benutzers und für die Autorisierung des Zugriffs auf bestimmte Ressourcen. Der angeforderte Dienst prüft die im Verzeichniseintrag des Benutzers enthaltenen benutzerspezifischen Konfigurationsangaben. Der Dienst verwendet diese Informationen, um die vom Benutzer angeforderten Aufgaben durchzuführen.

Die folgende Abbildung illustriert den Zugriff auf Benutzereinträge für die Durchführung der Benutzerauthentifizierung und -autorisierung hinsichtlich der Bereitstellung eines Dienstes für den Benutzer.

**Abbildung 3-1** Ein einzelner Benutzereintrag unterstützt viele Dienste

Eine der sich aus diesem System ergebenden Funktionen besteht darin, dass sich ein Benutzer im Web bei jedem Java ES-Dienst anmelden kann und dadurch automatisch bei den anderen Diensten des Systems authentifiziert wird. Diese leistungsstarke Funktion von Java Enterprise System ist unter der Bezeichnung *Single Sign-On* bekannt.

## Authentifizierung und Single Sign-On

Die Authentifizierungs- und Autorisierungsdienste von Java ES werden durch Access Manager bereitgestellt. Access Manager verwendet Informationen aus Directory Server, um als Broker für die Interaktion von Benutzern mit Java ES-Webdiensten und anderen webbasierten Diensten in einem Unternehmen zu fungieren.

Access Manager verwendet eine externe Komponente, den so genannten Richtlinienagenten. Der Richtlinienagent stellt eine Verbindung zu dem Webserver her, der als Host für einen Dienst bzw. eine Ressource fungiert, die durch Access Manager gesichert ist. Der Richtlinienagent schaltet sich bei Anforderungen des Benutzers im Namen von Access Manager in die gesicherten Ressourcen ein. Für einige Java ES-Komponenten, wie Portal Server und Communications Express, werden die Funktionen des Richtlinienagenten von einer Access Manager-Unterkomponente bereitgestellt (siehe „[Sun Java System Access Manager 6 2005Q1](#)“ auf Seite 82).

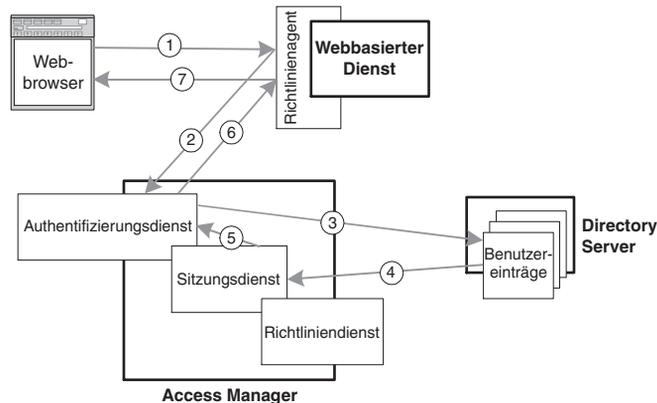
## Authentifizierung

Access Manager beinhaltet einen Authentifizierungsdienst zur Überprüfung der Identitäten der Benutzer, die (über HTTP oder HTTPS) Zugriff auf Webdienste innerhalb eines Unternehmens anfordern. Beispiel: Ein Firmenangestellter, der die Telefonnummer eines Kollegen nachschlagen muss, verwendet einen Browser, um zum Online-Telefonbuch des Unternehmens zu gelangen. Um sich beim Telefonbuchdienst anzumelden, muss der Benutzer eine Benutzer-ID und ein Passwort angeben.

Die Authentifizierungssequenz ist in [Abbildung 3-2](#) dargestellt. Ein Richtlinienagent schaltet sich in die Anfrage, die eine Anmeldung beim Telefonbuch enthält, ein (1) und schickt eine Anfrage an den Authentifizierungsdienst (2). Der Authentifizierungsdienst prüft die Benutzer-ID und das Kennwort anhand der in Directory Server gespeicherten Informationen (3). Wenn die Anmeldeanforderung gültig ist, wird der Benutzer authentifiziert (4, 5 und 6) und dem Mitarbeiter wird das Firmentelefonbuch angezeigt (7). Wenn die Anmeldeanforderung nicht gültig ist, wird eine Fehlermeldung generiert und die Authentifizierung schlägt fehl.

Der Authentifizierungsdienst unterstützt auch zertifikatbasierte Authentifizierungen über HTTPS.

**Abbildung 3-2** Authentifizierungssequenz



## Single Sign-On

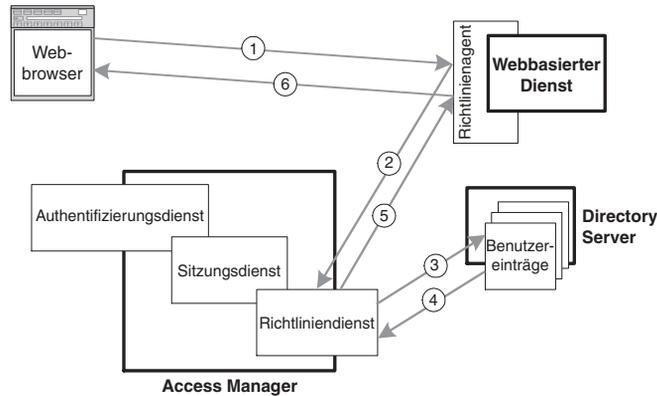
In dem in den vorherigen Abschnitten beschriebenen Authentifizierungsszenario fehlt ein wichtiger Schritt. Wenn die Authentifizierungsanfrage eines Benutzers überprüft ist, wird der Sitzungsdienst von Access Manager aufgerufen (4), wie in [Abbildung 3-2](#) dargestellt. Der Sitzungsdienst erstellt ein Sitzungs-Token, das Identitätsinformationen des Benutzers und eine Token-ID enthält (5). Das Sitzungs-Token wird an den Richtlinienagenten zurückgesendet (6), der es (als Cookie) an den Browser weiterleitet (7), von dem die Authentifizierungsanforderung ausging.

Wenn der authentifizierte Benutzer versucht, auf einen anderen gesicherten Dienst zuzugreifen, leitet der Browser das Sitzungs-Token an den entsprechenden Richtlinienagenten weiter. Der Richtlinienagent überprüft beim Sitzungsdienst, ob die frühere Authentifizierung des Benutzers noch gültig ist, und der Benutzer erhält Zugriff auf den zweiten Dienst, ohne dass er noch einmal eine Benutzer-ID und ein Passwort eingeben muss.

Ein Benutzer muss sich daher nur ein einziges Mal anmelden, um bei mehreren von Java Enterprise System bereitgestellten webbasierten Diensten authentifiziert zu werden. Die Single Sign-On-Authentifizierung bleibt gültig, bis sich der Benutzer explizit abmeldet oder die Sitzung abläuft.

## Autorisierung

Access Manager beinhaltet auch einen Richtliniendienst, mit dem der Zugriff auf webbasierte Ressourcen in einer Java ES-Umgebung gesteuert werden kann. Eine *Richtlinie* ist eine Regel, die angibt, wer autorisiert ist, unter bestimmten Bedingungen auf eine bestimmte Ressource zuzugreifen. Die Autorisierungssequenz ist in der folgenden Abbildung dargestellt.

**Abbildung 3-3** Autorisierungssequenz

Wenn ein authentifizierter Benutzer eine Anforderung für eine durch Access Manager gesicherte Ressource abschickt (1), benachrichtigt der Richtlinienagent den Richtliniendienst (2), der anhand der in Directory Server vorhandenen Informationen (3) die Zugriffsrichtlinie für die Ressource prüft, um festzustellen, ob der Benutzer über die erforderlichen Zugriffsberechtigungen verfügt (4). Wenn der Benutzer zugriffsberechtigt ist (5), wird die Ressourcenanfrage ausgeführt (6).

Access Manager bietet alle Mittel, die erforderlich sind, um innerhalb eines Unternehmens Richtlinien zu definieren, zu bearbeiten, zu gewähren, zu widerrufen und zu löschen. Die Richtlinien werden in Directory Server gespeichert und über richtlinienbezogene Attribute in Organisationseinträgen konfiguriert. Es können auch Rollen für Benutzer definiert und in Richtliniendefinitionen integriert werden.

Die Access Manager-Richtlinienagenten sind für die Durchsetzung der Richtlinien zuständig. Wenn der Richtliniendienst eine Zugriffsanforderung zurückweist, verhindert der Richtlinienagent, dass der betreffende Benutzer auf die gesicherten Ressourcen zugreift.

## In diesem Kapitel enthaltene Schlüsselbegriffe

Dieser Abschnitt erläutert in diesem Kapitel verwendete wichtige technische Begriffe, wobei der Schwerpunkt darauf liegt, die Beziehungen zwischen diesen Begriffen und ihrer Verwendung im Java Enterprise System-Kontext zu verdeutlichen.

**Verzeichnis** Eine spezielle Art von Datenbank, die für das Lesen – und weniger das Schreiben – von Daten optimiert ist. Die meisten Verzeichnisse beruhen auf LDAP (Lightweight Directory Access Protocol), einem als Industriestandard etablierten Protokoll.

**Richtlinie** Eine Regel, die angibt, wer autorisiert ist, unter bestimmten Bedingungen auf eine bestimmte Ressource zuzugreifen. Diese Regel kann auf Gruppen von Benutzern oder Rollen in einer Organisation basieren.

**Einzelidentität** Eine Identität, über die ein Benutzer dank eines einzelnen Benutzereintrags in einem Java Enterprise System-Verzeichnis verfügt. Anhand dieser einzelnen Benutzeridentität kann einem Benutzer der Zugriff auf verschiedene Java Enterprise System-Ressourcen gestattet werden, beispielsweise auf Portale, Webseiten und Dienste, wie Nachrichtendienste, Kalenderdienste und Instant Messaging.

**Single Sign-On** Eine Funktion, die es ermöglicht, die Authentifizierung eines Benutzers bei einem Dienst in einem verteilten System automatisch auf andere Dienste in diesem System anzuwenden.

# Aufgaben des Lösungslebenszyklus von Java Enterprise System

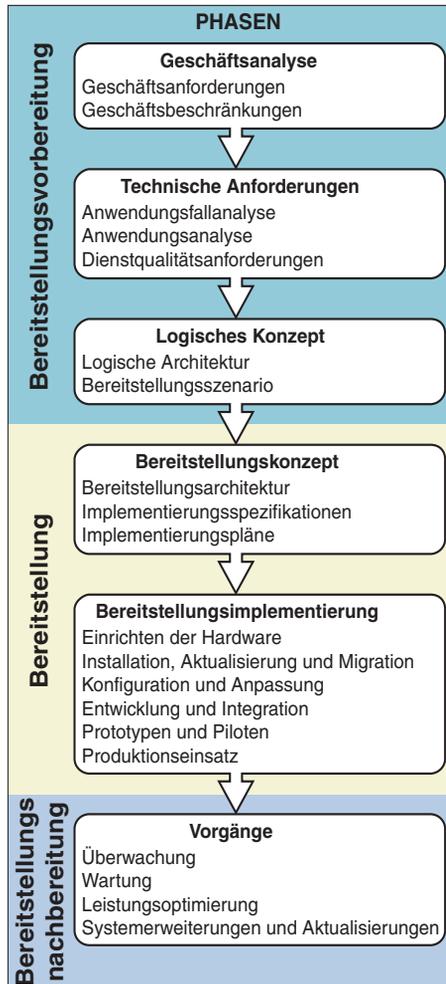
Der Java ES-Lösungslebenszyklus wurde im [Kapitel 1, „Einführung in Java Enterprise System“](#), als Standardansatz für die Implementierung von Geschäftslösungen mit der Java ES-Software vorgestellt. In diesem Kapitel werden die zu den einzelnen Lebenszyklusphasen gehörenden Aufgaben beschrieben. Das Lebenszyklusdiagramm wird zu Referenzwecken noch einmal wiederholt ([Abbildung 4-1 auf Seite 70](#)).

In diesem Kapitel werden die für jede Phase relevanten Konzepte und die zugehörige Terminologie besprochen. Der Schwerpunkt liegt dabei auf den Bereitstellungsaufgaben, insbesondere auf dem Bereitstellungskonzept und den Aufgaben der Bereitstellungsimplementierung.

Das Kapitel ist nach den folgenden drei Lebenszyklusphasen aufgebaut.

- [„Bereitstellungsvorbereitung“ auf Seite 71](#)
- [„Bereitstellung“ auf Seite 72](#)
- [„Bereitstellungsnachbereitung“ auf Seite 78](#)
- [„In diesem Kapitel enthaltene Schlüsselbegriffe“ auf Seite 79](#)

**Abbildung 4-1** Aufgaben des Lösungslebenszyklus



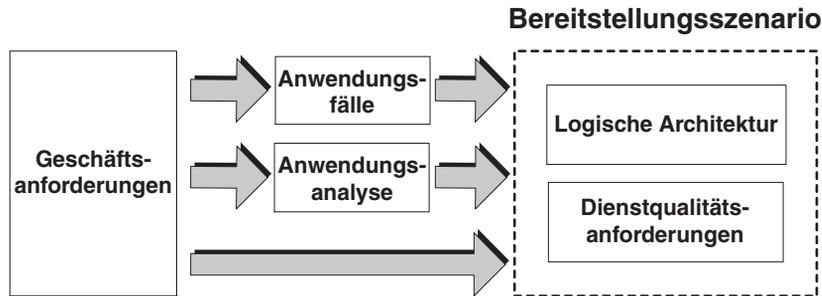
# Bereitstellungsvorbereitung

In der Lebenszyklusphase *Bereitstellungsvorbereitung* setzen Sie eine Analyse der Geschäftsanforderungen in ein *Bereitstellungsszenario* um. Das Bereitstellungsszenario dient als Spezifikation für ein Bereitstellungskonzept.

Die vor der Bereitstellung notwendigen Aufgaben werden zu drei Phasen zusammengefasst, die in *Abbildung 4-1* dargestellt sind:

- **Geschäftsanalyse:** In dieser Phase definieren Sie die Geschäftsziele eines Bereitstellungsvorschlags und nennen die Geschäftsanforderungen und -beschränkungen, die zur Erreichung dieses Ziels erfüllt werden müssen.
- **Technische Anforderungen:** In dieser Phase erstellen Sie anhand der Ergebnisse der Geschäftsanalyse *Anwendungsfälle*, die die Interaktion der Benutzer mit einem vorweggenommenen Softwaresystem abbilden. Außerdem legen Sie die für diese Anwendungsfälle erwarteten Anwendungsmuster fest. Unter Berücksichtigung der Geschäftsanalyse und der Anwendungsanalyse formulieren Sie Dienstqualitätsanforderungen (siehe *Tabelle 2-2 auf Seite 48*), die der Bereitstellungsvorschlag einhalten muss.
- **Logisches Konzept:** In dieser Phase analysieren Sie die in der Phase der technischen Anforderungen entwickelten Anwendungsfälle, um festzustellen, welche Java ES-Infrastrukturkomponenten und welche kundenspezifisch entwickelten Anwendungskomponenten Sie für die Endbenutzer benötigen. Anhand der im *Kapitel 2, „Lösungsarchitekturen von Java Enterprise System“*, besprochenen Konzepte entwerfen Sie eine logische Architektur. Die logische Architektur enthält alle Komponenten und alle zwischen den Komponenten stattfindenden Interaktionen, die notwendig sind, damit die Anwendungsfälle einer bestimmten Softwarelösung umgesetzt werden.

Die logische Architektur wird, wie in der nachfolgenden Abbildung dargestellt, zusammen mit Leistungs-, Verfügbarkeits-, Sicherheits- und anderen Dienstqualitätsanforderungen im Bereitstellungsszenario zusammengefasst. Weitere Informationen über die vor der Bereitstellung anfallenden Lebenszyklusphasen finden Sie im *Java Enterprise System Deployment Planning Guide*.

**Abbildung 4-2** Spezifizieren eines Bereitstellungsszenarios

## Bereitstellung

In den *Bereitstellung*phasen des Lebenszyklus setzen Sie ein Bereitstellungsszenario in ein Bereitstellungs-konzept um, das Sie dann implementieren, testen und schließlich in einer Produktionsumgebung einsetzen.

Der Bereitstellungsprozess umfasst im Allgemeinen Softwarekomponenten aller Schichten und aller Infrastrukturdienstebenen, die zur Unterstützung einer Softwarelösung erforderlich sind. Meist müssen Sie sowohl kundenspezifisch entwickelte Anwendungskomponenten (J2EE-Komponenten, Webdienste oder andere Server) sowie die für die Lösung benötigten Java ES-Komponenten bereitstellen.

Die Bereitstellungsaufgaben werden zu drei Phasen zusammengefasst, die *Abbildung 4-1* darstellt:

- **Bereitstellungskonzept.** Das Bereitstellungskonzept hängt sowohl von der logischen Architektur einer Lösung als auch von der Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit, Zweckmäßigkeit und anderen Dienstqualitätsanforderungen ab, die eine Lösung erfüllen muss. Die Dimension der Dienstqualität einer Bereitstellungsarchitektur spielt in der Phase des Bereitstellungskonzepts eine wichtige Rolle.
- **Bereitstellungsimplementierung.** Die Implementierung eines Bereitstellungskonzepts stellt einen häufig iterativ verlaufenden Prozess dar, der die Einrichtung der Hardware, die Installation und Konfiguration der Software, die Entwicklung und Integration, das Testen und andere Aspekte des Produktionseinsatzes umfasst.

Diese beiden Phasen des Bereitstellungsprozesses werden in den folgenden Abschnitten eingehender beschrieben.

## Bereitstellungskonzept

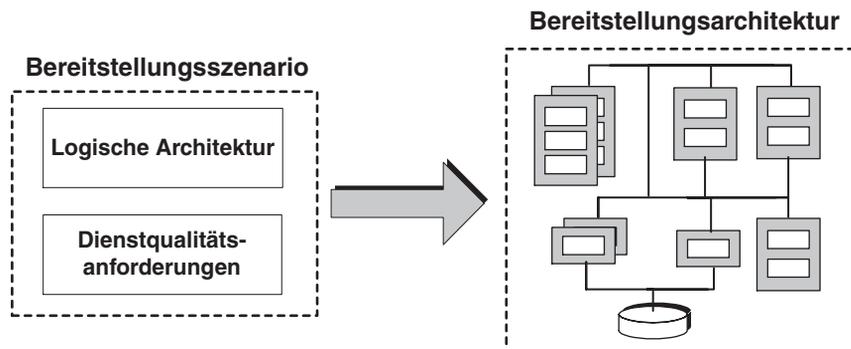
In der Phase des Bereitstellungskonzepts erstellen Sie eine Bereitstellungsarchitektur auf hoher Ebene, gefolgt von Implementierungsspezifikationen auf niedriger Ebene.

### Bereitstellungsarchitektur

Eine Bereitstellungsarchitektur wird durch Zuordnung der logischen Modulblöcke einer Anwendung (die logische Architektur) zu einer physischen Computerumgebung erstellt. Hierbei müssen die im Bereitstellungsszenario festgelegten Dienstqualitätsanforderungen erfüllt sein.

Das Bereitstellungsszenario wird, wie in der folgenden Abbildung dargestellt, in eine Bereitstellungsarchitektur umgesetzt.

**Abbildung 4-3** Umsetzen eines Bereitstellungsszenarios in eine Bereitstellungsarchitektur



Ein Aspekt dieses Architekturkonzepts bildet die Festlegung der Größe der physischen Umgebung (Bestimmung der Anzahl der Computer und Schätzung ihrer Prozessorstärke und RAM-Anforderungen), die notwendig ist, um die Anforderungen hinsichtlich der Leistung, Verfügbarkeit, Sicherheit und anderer Dienstqualitätsanforderungen zu erfüllen. Nachdem Sie die Größe festgelegt haben, weisen Sie den Computern in der physischen Umgebung Java ES-Komponenten und Anwendungskomponenten zu. Die resultierende Bereitstellungsarchitektur muss die Funktionen der verschiedenen Computer, die Eigenschaften der Systeminfrastrukturdienste und die Beschränkungen für die Betriebs- oder Verfügbarkeitskosten berücksichtigen.

Je größer die Anzahl der Java ES-Komponenten im Bereitstellungsszenario ist und je höher die Dienstqualitätsanforderungen sind, desto höher sind die Ansprüche, die an die Computer und die Netzwerkbandbreite gestellt werden. Wenn die Hardware begrenzt oder extrem teuer ist, müssen Sie eventuell Kompromisse zwischen den Festkosten (Hardware) und den variablen Kosten (Personalbedarf) oder zwischen den verschiedenen Dienstqualitätsanforderungen schließen. Möglicherweise müssen Sie aber auch Ihr Konzept vereinfachen.

Die Konzeption einer Bereitstellungsarchitektur erfolgt oft in sich wiederholenden Schritten. Als Ausgangspunkt für ein Bereitstellungskonzept entwickelt das Java Enterprise System jedoch eine Reihe von *Referenz-Bereitstellungsarchitekturen*.

Eine Referenzarchitektur basiert auf einem bestimmten Bereitstellungsszenario: Einer logischen Architektur mit bestimmten Dienstqualitätsanforderungen. In der Referenzarchitektur wird eine Softwarelösung so innerhalb einer bestimmten physischen Umgebung bereitgestellt, dass die vorgegebenen Dienstqualitätsanforderungen erfüllt werden. Das Testen der Leistung bei vorgegebenen Lasten basiert auf derselben Gruppe von Anwendungsfällen, aus denen das Bereitstellungsszenario entwickelt wurde. Dokumentation zur Referenzarchitektur ist für Java ES-Kunden öffentlich verfügbar.

Auf der Grundlage einer Referenz-Bereitstellungsarchitektur oder einer Kombination von Referenzarchitekturen können Sie eine erste näherungsweise Bereitstellungsarchitektur entwerfen, die Ihre eigenen Bereitstellungsszenario-Anforderungen erfüllt. Sie können unter Berücksichtigung der Unterschiede zwischen Ihrem eigenen Bereitstellungsszenario und den Bereitstellungsszenarios, auf denen die Referenzarchitekturen basieren, die Referenzarchitekturen anpassen oder als Referenzpunkte verwenden. Auf diese Weise können Sie die Auswirkungen Ihrer eigenen Größen-, Leistungs-, Sicherheits-, Verfügbarkeits-, Kapazitäts- und Zweckmäßigkeitserfordernisse beurteilen.

## Implementierungsspezifikationen

Implementierungsspezifikationen enthalten die für die Implementierung einer Bereitstellungsarchitektur notwendigen Einzelheiten. Die Spezifikationen enthalten in der Regel folgende Informationen:

- Vorhandene Hardware, mit Computern, Speichergeräten, Lastausgleichsmodulen und Netzkabeln
- Betriebssysteme
- Netzwerkstruktur, mit Subnetzen und Sicherheitszonen

- Einzelheiten des Verfügbarkeitskonzepts
- Einzelheiten des Sicherheitskonzepts
- Für das Einrichten von Endbenutzern benötigte Informationen über das Verzeichniskonzept

## Implementierungspläne

Implementierungspläne beschreiben, wie Sie die Durchführung der verschiedenen Aufgaben der Phase der Bereitstellungsimplementierung geplant haben. Die Pläne enthalten in der Regel folgende Aufgaben

- Einrichten der Hardware
- Installation, Aktualisierung und Migration der Software
- Konfiguration und Anpassung des Systems
- Entwicklung und Integration
- Tests
- Produktionseinsatz

## Bereitstellungsimplementierung

Die Implementierung eines Bereitstellungskonzepts besteht aus den im vorherigen Abschnitt aufgeführten Aufgaben, die auch in [Abbildung 4-1](#) dargestellt sind. Die Reihenfolge dieser Aufgaben ist nicht streng vorgegeben, da der Bereitstellungsprozess als solcher iterativ verläuft. In den nachfolgenden Unterabschnitten werden die wichtigsten Aufgaben der Bereitstellungsimplementierung in der Reihenfolge beschrieben, in der sie in der Regel ausgeführt werden. Detaillierte Dokumentation zu diesen Aufgaben entnehmen Sie der *Java Enterprise System Documentation Roadmap*.

### Einrichten der Hardware

Die Implementierungsspezifikationen enthalten alle Details Ihrer physischen Umgebung: Computer, Netzwerkstruktur, Netzwerkhardware (mit Leitungen, Schaltern, Routern und Lastausgleichsmodulen), Speichergeräte usw. Die gesamte Hardware muss als Plattform für Ihre Java ES-Lösung eingerichtet werden.

## Installation, Aktualisierung und Migration der Software

Aus der Bereitstellungsarchitektur und den in den Implementierungsspezifikationen enthaltenen zusätzlichen Details erfahren Sie, welche Anwendungskomponenten und welche Java ES-Komponenten auf jedem Computer Ihrer physischen Umgebung vorhanden sein müssen. Mit dem in Java ES integrierten Installationsprogramm installieren Sie auf jedem Computer Ihrer Bereitstellungsarchitektur die entsprechenden Java ES-Komponenten (siehe [„Das integrierte Installationsprogramm von Java Enterprise System“](#) auf Seite 60).

Ihr Installationsplan beschreibt den Ablauf und den Umfang der Installationssitzungen. Nach welchem Ansatz Sie die Installationen durchführen, kann jedoch davon abhängen, ob Sie eine Neuinstallation von Java Enterprise System durchführen oder ob Sie zuvor bereits installierte Java ES-Komponenten aktualisieren oder ob Sie Komponenten anderer Hersteller durch Java Enterprise System ersetzen. Bei den beiden letztgenannten Java ES-Einführungsszenarien ist es häufig notwendig, dass aus Gründen der Kompatibilität Daten oder Anwendungscode migriert werden müssen.

## Konfiguration und Anpassung des Systems

Damit die verschiedenen Systemkomponenten als integriertes System zusammenarbeiten, muss die Systemkonfiguration an einigen Stellen angepasst werden. Dazu gehört zuerst die für den Start jeder Systemkomponente notwendige Erstkonfiguration. Anschließend müssen alle Java ES-Komponenten so konfiguriert werden, dass sie mit den Komponenten kommunizieren, mit denen sie interagieren.

Entsprechend der für jede Komponente vorliegenden Verfügbarkeitslösung muss darüber hinaus die Hochverfügbarkeit konfiguriert werden. Die Benutzer müssen eingerichtet werden, damit sie auf die verschiedenen Dienste zugreifen können. Darüber hinaus müssen die Richtlinien und die Steuerung für die Authentifizierung und die Autorisierung eingerichtet werden (siehe [„Integrierte Identitäts- und Sicherheitsdienste“](#) auf Seite 62).

In den meisten Fällen gehören zu den Konfigurationsaufgaben auch gewisse Anpassungen der Java ES-Komponenten, damit exakt die benötigten Funktionen verfügbar sind. So passen Sie in der Regel den Portal Server an, damit er Zugangskanäle bereitstellt, den Access Manager, damit er Autorisierungsaufgaben durchführt und den Messaging Server, damit er die Virusprüfung und den Spam-Filter einsetzt.

## Entwicklung und Integration

Die logische Architektur, die im Bereitstellungsszenario festgelegt ist, bestimmt das Ausmaß der kundenspezifischen *Entwicklungsarbeit*, die zur Implementierung einer Lösung erforderlich ist.

Für einige Bereitstellungen kann die Entwicklungsarbeit viel umfangreicher sein. Möglicherweise müssen Sie unter Verwendung von J2EE-Komponenten, die in einer Application Server- oder Web Server-Umgebung ausgeführt werden, ganz neue Geschäfts- und Präsentationsdienste entwickeln. In diesen Fällen erstellen Sie einen Prototyp für die Lösung und testen das Konzept, bevor Sie mit der vollständigen Entwicklung anfangen.

Für Lösungen, bei denen umfangreiche Entwicklungsarbeiten notwendig sind, bietet Sun Java Studio verschiedene Tools für die Programmierung verteilter Komponenten oder Geschäftsdienste. Sun Java Studio vereinfacht die Programmierung und das Testen von Anwendungen, die von der Java ES-Infrastruktur unterstützt werden.

In bestimmten Situationen müssen Java ES-Komponenten mit Legacy-Anwendungen oder Diensten anderer Hersteller integriert werden. Eine solche Integration kann in der Datenschicht vorhandene Verzeichnis- oder Datendienste oder in der Geschäftsdienstschicht vorhandene Komponenten betreffen. Für die Integration von Java ES-Komponenten in solche Systeme kann die Migration von Daten oder Anwendungscode notwendig sein.

Die J2EE-Plattform bietet ein Connector Framework, mit dem Sie vorhandene Anwendungen in die Application Server-Umgebung aufnehmen, indem Sie J2EE-Ressourcenadapter entwickeln. Auch Message Queue bietet für die Integration diverser Anwendungen eine robuste, asynchrone Messaging-Funktion.

## Testen von Prototypen und Piloten

Abhängig vom Umfang der notwendigen Anpassungs- oder Entwicklungsarbeiten, müssen Sie Ihre Bereitstellungsarchitektur an bestimmten Punkten überprüfen. Sie müssen die Lösung im Hinblick auf die Anwendungsfälle testen und prüfen, ob die Dienstqualitätsanforderungen erfüllt werden.

Wenn Sie relativ wenig kundenspezifisch entwickelte Dienste besitzen (die Bereitstellung also überwiegend Out-of-the-Box erfolgt), müssen für Ihre Lösung lediglich die Java ES-Komponenten angepasst und ein Pilottest des Systems durchgeführt werden.

Wenn Sie jedoch eine wichtige neue Anwendungslogik entwickelt und benutzerdefinierte Dienste erstellt haben, müssen Sie wahrscheinlich umfangreichere Tests durchführen, wie Prototyp-Tests, Integrationstest usw.

Wenn diese Tests Schwachstellen Ihrer Bereitstellungsarchitektur aufzeigen, müssen Sie die Architektur ändern und erneut testen. Dieser schrittweise Prozess sollte schließlich zu einer Bereitstellungsarchitektur und einer Implementierung führen, die die Bereitstellung in einer Produktionsumgebung gewährleisten kann.

## Produktionseinsatz

Der Produktionseinsatz besteht in der Anwendung Ihrer Bereitstellungsimplementierung in einer Produktionsumgebung. Diese Phase umfasst das Installieren, Konfigurieren und Starten der verteilten Anwendungen und Infrastrukturdienste in einer Produktionsumgebung, die Einrichtung von Produktionssystem-Endbenutzern sowie die Einrichtung von Single Sign-On, Zugriffsrichtlinien usw. In der Regel beginnen Sie mit einer eingeschränkten Bereitstellung und dehnen diese dann auf das gesamte Unternehmen aus. Hierbei führen Sie Testläufe aus, in denen Sie die Belastung stetig steigern, um zu bestätigen, dass die Qualitätsanforderungen erfüllt werden.

# Bereitstellungsnachbereitung

In der Lebenszyklusphase *Bereitstellungsnachbereitung* betreiben Sie eine bereitgestellte Lösung in einer Produktionsumgebung. Zur Betriebsphase des Lebenszyklus gehörende folgende Aufgaben:

- **Überwachung:** Zu dieser Aufgabe gehört die regelmäßige Überwachung der Systemleistung und der Systemfunktionen.
- **Wartung:** Zu dieser Aufgabe gehören die täglich anfallenden Verwaltungsfunktionen, wie das Hinzufügen neuer Benutzer zu einem System, das Ändern von Passwörtern, das Hinzufügen neuer administrativer Benutzer, das Ändern von Zugriffsberechtigungen, die Durchführung regelmäßiger Sicherungen usw.
- **Leistungsoptimierung:** Zu dieser Aufgabe gehört die Auswertung regelmäßiger Überwachungsinformationen, um Engpässe beim Systembetrieb zu erkennen und diese dann durch Änderungen der Konfigurationseinstellungen, Hinzufügen von Kapazität usw. zu beseitigen.

- Systemerweiterungen und Aktualisierungen: Zu diesen Aufgaben gehört das Hinzufügen neuer Java ES-Komponenten zu einem System, um neue Funktionen verfügbar zu machen, oder um Nicht-Java ES-Komponenten zu ersetzen. In beiden Fällen kann eine Neukonzeption des Systems notwendig sein, die mit den ersten Phasen des Lösungslebenszyklus anfängt. Aktualisierungsaufgaben sind weniger umfangreich, da es sich meist um die Aktualisierung von Java ES-Komponenten handelt.

Jede Java ES-Komponente verfügt über eigene Tools für die Konfiguration, Optimierung oder Verwaltung. Ziel ist die Bereitstellung einer gemeinsamen Überwachungs- und Verwaltungsinfrastruktur sowie von Tools, mit denen das System als Ganzes verwaltet werden kann.

## In diesem Kapitel enthaltene Schlüsselbegriffe

Dieser Abschnitt erläutert in diesem Kapitel verwendete wichtige technische Begriffe, wobei der Schwerpunkt darauf liegt zu verdeutlichen, welche Beziehung zwischen diesen Begriffen im Java Enterprise System-Kontext vorliegt.

**Bereitstellung** Ein Abschnitt im Lebenszyklus einer Java Enterprise System-Lösung, in dem ein Bereitstellungsszenario in ein Bereitstellungs-konzept überführt, dann implementiert, als Prototyp getestet und schließlich in einer Produktionsumgebung eingesetzt wird. Das Endprodukt dieses Prozesses wird ebenfalls als Bereitstellung (oder bereitgestellte Lösung) bezeichnet.

**Bereitstellungsszenario** Die *logische Architektur* einer Java Enterprise System-Lösung und die Dienstqualitätsanforderungen, die die Lösung erfüllen muss, um den Geschäftsanforderungen zu entsprechen. Die Dienstqualitätsanforderungen betreffen u. a. Leistung, Verfügbarkeit, Sicherheit, Zweckmäßigkeit und Skalierbarkeit/latente Kapazität. Ein Bereitstellungsszenario ist der Ausgangspunkt für ein Bereitstellungs-konzept.

**Entwicklung** Eine Aufgabe im Bereitstellungsprozess der Java Enterprise System-Lösung, bei der die angepassten Komponenten einer *Bereitstellungsarchitektur* programmiert und getestet werden.

**Bereitstellungsvorbereitung** Ein Abschnitt im Lebenszyklus der Java Enterprise System-Lösung, in dem die Geschäftsanforderungen in ein *Bereitstellungsszenario* umgesetzt werden: Eine *logische Architektur* und eine Reihe von Dienstqualitätsanforderungen, die die Lösung erfüllen muss.

**Bereitstellungsnachbereitung** Ein Abschnitt im Lebenszyklus einer Java Enterprise System-Lösung, in dem verteilte Anwendungen gestartet, überwacht, zur Optimierung der Leistung angepasst und dynamisch mit neuen Funktionen aufgerüstet werden.

**Referenz-Bereitstellungsarchitektur** Eine *Bereitstellungsarchitektur*, die hinsichtlich der Leistung entworfen, implementiert und getestet wurde. Referenz-Bereitstellungsarchitekturen dienen als Ausgangspunkt für die Konzeption maßgeschneiderter Lösungen.

**Anwendungsfall** Eine bestimmte Endbenutzeraufgabe oder eine Reihe von Aufgaben, die eine *Verteilte Unternehmensanwendung* ausführt und die als Basis für die Konzipierung, das Testen und das Messen der Leistung der Anwendung dient.

# Referenzliste: Komponenten von Java Enterprise System

Dieser Anhang enthält eine Referenzliste aller Java ES-Komponenten, die in die folgenden Kategorien unterteilt ist:

- **Beschreibungen der Systemdienstkomponenten.** Diese Komponenten bieten wesentliche Java ES-Infrastrukturdienste, die für die Unterstützung verteilter Unternehmensanwendungen benötigt werden. Zu diesen unter „Wozu benötigen Sie Java Enterprise System?“ auf Seite 18 beschriebenen Diensten gehören Zugangsdienste, Kommunikations- und Zusammenarbeitsdienste, Identitäts- und Sicherheitsdienste, Web- und Anwendungsdienste sowie Verfügbarkeitsdienste.
- **Beschreibung der Dienstqualitätskomponenten.** Diese Komponenten werden verwendet, um die Dienstqualitäten von Systemdienstkomponenten oder verteilten Anwendungskomponenten zu verbessern. Bei einigen handelt es sich um Verfügbarkeitskomponenten, die für einen nahezu kontinuierlichen Betrieb des Systems eingesetzt werden. Bei anderen handelt es sich um Zugriffskomponenten, die den gesicherten Zugriff von Endbenutzern auf Systemdienste unterstützen. Oder es handelt sich um Systemverwaltungskomponenten, mit denen die Zweckmäßigkeit der Java ES-Lösungen verbessert wird.
- **Gemeinsam genutzte Komponenten.** Diese Komponenten sind lokale Bibliotheken, die von allen Java ES-Komponenten, die auf bestimmten Hostcomputern laufen, gemeinsam genutzt werden können.

In diesem Anhang sind die Java ES-Komponenten innerhalb ihrer jeweiligen Kategorien und Unterkategorien alphabetisch aufgelistet.

Einen Wegweiser zur Dokumentation der einzelnen Komponenten finden Sie hier: *Java Enterprise System Documentation Roadmap* (<http://docs.sun.com/doc/819-0055>).

# Beschreibungen der Systemdienstkomponenten

Java ES-Systemdienstkomponenten bieten die Infrastrukturdienste, die für die Unterstützung verteilter Unternehmensanwendungen benötigt werden. Die Java ES-Systemdienstkomponenten werden in den folgenden Abschnitten beschrieben:

- [Sun Java System Access Manager 6 2005Q1](#)
- [Sun Java System Application Server Enterprise Edition 8 2005Q1](#)
- [Sun Java System Calendar Server 6 2005Q1](#)
- [Sun Java System Directory Server 5 2005Q1](#)
- [Sun Java System Instant Messaging 7 2005Q1](#)
- [Sun Java System Message Queue 3 2005Q1](#)
- [Sun Java System Messaging Server 6 2005Q1](#)
- [Sun Java System Portal Server 6 2005Q1](#)
- [Sun Java System Web Server 6.1 2005Q1](#)

## Sun Java System Access Manager 6 2005Q1

Sun Java System Access Manager (Access Manager) stellt für eine Organisation eine Infrastruktur zur Verwaltung der Vorgänge zur Verfügung, mit denen die digitalen Identitäten von Kunden, Mitarbeitern und Partnern verwaltet werden, die mit den webbasierten und nicht webbasierten Anwendungen arbeiten. Da diese Ressourcen möglicherweise weiträumig über interne und externe Computernetzwerke hinweg verstreut sind, werden Attribute, Richtlinien und Berechtigungen definiert und auf jede Identität angewendet, um den Zugriff auf diese Technologien zu verwalten.

Das Java ES-Installationsprogramm stellt Access Manager als installierbare Einzelkomponente zur Verfügung. Bei Bedarf können die folgenden Access Manager-Unterkomponenten separat installiert werden:

- **Identity Management and Policy Services Core:** Stellt die Mittel zum Erstellen und Verwalten von Benutzeridentitäten sowie zum Definieren und Auswerten der Richtlinien zur Verfügung, die auf Basis der Benutzeridentitäten den Zugriff auf die Java ES-Ressourcen ermöglichen. Zu diesen Unterkomponenten gehören auch Access Manager SDK und der Delegated Administrator (siehe „[Sun Java System Delegated Administrator 6 2005Q1](#)“ auf Seite 91).

- **Access Manager SDK:** Bietet eine Remote-Schnittstelle zu Access Manager. Diese Unterkomponente muss auf allen Computern installiert werden, auf denen sich eine Java ES-Komponente befindet, die remote auf Access Manager zugreift.
- **Access Manager Administration Console:** Diese grafische Benutzeroberfläche fasst die Identitätsdienste und die Richtlinienverwaltung zu einer einzelnen Oberfläche zusammen, die Benutzern das Erstellen und Verwalten von Benutzerkonten, Dienstattributen und Zugangsregeln in Directory Server ermöglicht.
- **Common Domain Services for Federation Management:** Ermöglicht Benutzern die Verwendung einer einzelnen Identität, um auf Anwendungen zuzugreifen, die von mehreren angegliederten Dienstanbietern angeboten werden.

## Sun Java System Application Server Enterprise Edition 8 2005Q1

Sun Java System Application Server (Application Server) bietet eine J2EE-kompatible Plattform für die Entwicklung und Bereitstellung von Anwendungs- und Webdiensten. Application Server liefert die Infrastruktur-Services für die Interaktion zwischen eng gekoppelten verteilten Komponenten, einschließlich des Aufrufs über die Remote-Methode und weitere Laufzeitdienste.

Das Java ES-Installationsprogramm stellt Application Server als installierbare Einzelkomponente zur Verfügung. Bei Bedarf können die folgenden Application Server-Unterkomponenten separat installiert werden:

- **Domain Administration Server:** Bietet serverseitige Verwaltungsfunktionen, wie die Verwaltung und Konfiguration von Application Server und die Bereitstellung von J2EE-Komponenten und -Anwendungen.
- **Application Server Administration Client:** Bietet grafische Verwaltungs-Clients, mit denen Sie Application Server-Installationen und gehostete Anwendungen verwalten und konfigurieren können. Der Administration Client hilft auch bei der Bereitstellung von Anwendungen.
- **Command Line Administration Tool:** Bietet über Befehlszeilen arbeitende Verwaltungs-Clients, mit denen Sie Application Server-Installationen und gehostete Anwendungen verwalten und konfigurieren können. Das Tool hilft auch bei der Bereitstellung von Anwendungen.

- **Load Balancing Plug-in:**
- **PointBase:** Bietet eine eingebettete Datenbank, die für persistente Vorgänge genutzt werden kann
- **Beispielanwendungen:**

## Sun Java System Calendar Server 6 2005Q1

Sun Java System Calendar Server (Calendar Server) ist eine skalierbare, webbasierte Lösung, die für zentrale Kalender- und Planungsfunktionen in Unternehmen und bei Dienst Anbietern verwendet wird. Calendar Server unterstützt persönliche Kalender und Gruppenkalender sowie Kalender für Ressourcen, wie Konferenzräume und Geräte.

Das Java ES-Installationsprogramm stellt Calendar Server als installierbare Einzelkomponente zur Verfügung.

## Sun Java System Directory Server 5 2005Q1

Sun Java System Directory Server (Directory Server) bietet einen zentralen Verzeichnisdienst für Ihre Intranet-, Netzwerk- und Extranet-Informationen. Directory Server integriert sich in vorhandene Systeme und arbeitet als zentrales Repository für die Konsolidierung von Mitarbeiter-, Kunden-, Lieferanten- und Partnerinformationen. Sie können Directory Server dahin gehend erweitern, dass die Verwaltung von Benutzerprofilen und Voreinstellungen sowie die Extranet-Benutzerauthentifizierung ermöglicht werden.

Das Java ES-Installationsprogramm stellt Directory Server als installierbare Einzelkomponente zur Verfügung.

## Sun Java System Instant Messaging 7 2005Q1

Sun Java System Instant Messaging (Instant Messaging) ermöglicht Endbenutzern die Teilnahme an Instant Messaging- und Chat-Sitzungen, das gegenseitige Zusenden von Warnmeldungen und den sofortigen Austausch von Gruppenmitteilungen. Instant Messaging ist sowohl für Intranets als auch für das Internet geeignet und unterstützt die Interaktion mit anderen Anbietern von Instant Messaging.

Das Java ES-Installationsprogramm stellt Instant Messaging als installierbare Einzelkomponente zur Verfügung. Die folgenden Instant Messaging-Unterkomponenten können separat installiert werden:

- **Instant Messaging Server Core:** Enthält Server- und Multiplexer-Software.
- **Instant Messaging Ressourcen:**
- **Access Manager Instant Messaging-Dienst.**

## Sun Java System Message Queue 3 2005Q1

Sun Java System Message Queue (Message Queue) ist eine auf Standards basierende Lösung für das Problem der Kommunikation zwischen Anwendungen und für die zuverlässige Nachrichtenzustellung. Message Queue ist ein Messaging-System für Unternehmen, das den offenen Java Message Service (JMS) Standard implementiert.

Message Queue ist jedoch nicht nur ein JMS-Anbieter, sondern verfügt auch über Funktionen, die über die Mindestanforderungen der JMS-Spezifikation hinausgehen. Über die Message Queue-Software können Vorgänge, die auf unterschiedlichen Plattformen und unter unterschiedlichen Betriebssystemen ausgeführt werden, eine Verbindung mit einem allgemeinen Message Queue-Dienst herstellen, um Informationen zu senden und zu empfangen. Anwendungsentwickler können sich also auf die Geschäftslogik ihrer Anwendungen konzentrieren und müssen sich nicht mit den wenig interessanten Details der netzwerkübergreifenden Anwendungskommunikation befassen.

Message Queue ist in zwei Editions erhältlich:

- **Enterprise Edition** (Standardversion): Sie bietet Unterstützung für Nachrichtendienste mit mehreren Brokern, HTTP/HTTPS-Verbindungen, sichere und skalierbare Verbindungen, Failover für Client-Verbindungen und Client-Unterstützung für die Sprache C. Diese Edition ist für die Bereitstellung und Ausführung von Messaging-Anwendungen in einer umfangreichen Produktionsumgebung am besten geeignet.
- **Platform Edition:** Stellt grundlegende JMS-Unterstützung zur Verfügung und ist am besten für kleinere Bereitstellungs- und Entwicklungsumgebungen geeignet.

Das Java ES-Installationsprogramm stellt Message Queue Enterprise Edition und Message Queue Platform Edition als separat installierbare Komponenten zur Verfügung.

## Sun Java System Messaging Server 6 2005Q1

Sun Java System Messaging Server (Messaging Server) ist ein leistungsstarker, auf Industriestandards basierender Internet-Messaging-Server für Unternehmen und Dienstleister. Messaging Server ist für die zuverlässige Verarbeitung von Nachrichten in großem Umfang konzipiert und umfasst mehrere modulare, unabhängig konfigurierbare Komponenten, die Unterstützung für verschiedene E-Mail-Protokolle bieten.

Das Java ES-Installationsprogramm stellt Messaging Server als installierbare Einzelkomponente zur Verfügung. Nach der Installation kann jedoch jede Messaging Server-Instanz so konfiguriert werden, dass sie eine beliebige Anzahl verschiedener Messaging-Dienste bereitstellt, die aus der folgenden Gruppe von Messaging Server-Unterkomponenten stammen:

- **Message Store:** Ermöglicht das Speichern und Abrufen von Nachrichten.
- **Message Transfer Agent (MTA):** Unterstützt das Senden von E-Mails durch die Abwicklung von SMTP-Verbindungen, das Routing von E-Mails und das Ausliefern der Nachrichten an die entsprechenden Message-Speicher. Kann so konfiguriert werden, dass E-Mails an interne Speicher (Inbound) oder an externe E-Mailspeicher (Outbound) geliefert werden.
- **Message Multiplexor (MMP):** Unterstützt das Abrufen von E-Mails, indem der Zugriff auf den Message-Speicher (oder eine Gruppe von Speichern) für E-Mail-Clients realisiert wird, die IMAP- oder POP-Protokolle nutzen.
- **Message Express Multiplexor (MEM):** Unterstützt das Abrufen und Senden von E-Mails durch webbasierte (HTML) E-Mail-Clients.

## Sun Java System Portal Server 6 2005Q1

Sun Java System Portal Server (Portal Server) ist eine Portal Server-Lösung mit Identitätsaktivierung. Portal Server kombiniert wichtige Zugangsdienste, wie Personalisierung, Aggregation, Sicherheit, Integration und Suche. Mobile Access, eine Unterkomponente von Portal Server, realisiert den drahtlosen Zugang auf Portal Server für mobile Geräte, wie Handys und PDAs.

Das Java ES-Installationsprogramm stellt Portal Server als installierbare Einzelkomponente zur Verfügung.

## Sun Java System Web Server 6.1 2005Q1

Sun Java System Web Server (Web Server) ist ein sicherer Webserver mit mehreren Prozessen und Threads, der auf offenen Standards basiert. Web Server bietet eine hohe Leistung, Zuverlässigkeit, Skalierbarkeit und Verwaltbarkeit für Unternehmen jeder Größe. Web Server unterstützt eine große Reihe von Websoftware-Standards, einschließlich JDK 1.4.1, Java Servlet 2.3, JavaServer Pages™ (JSP™) 1.2, HTTP/1.1, PKCS #11, FIPS-140, 168-Bit Step-up-Zertifikate und verschiedene andere sicherheitsbasierte Standards.

Das Java ES-Installationsprogramm stellt Web Server als installierbare Einzelkomponente zur Verfügung.

## Beschreibung der Dienstqualitätskomponenten

Die für die Unterstützung der Java ES-Dienstkomponenten eingesetzten Komponenten sind in folgende Kategorien unterteilt:

- Verfügbarkeitskomponenten
- Zugriffskomponenten
- Verwaltungskomponenten

Die in diesen Kategorien enthaltenen Komponenten sind in den folgenden Abschnitten beschrieben.

## Beschreibung der Verfügbarkeitskomponenten

Verfügbarkeitskomponenten sorgen für eine nahezu kontinuierliche Systembetriebszeit der Systemdienstkomponenten und Anwendungskomponenten. In diesem Abschnitt werden die folgenden Java ES-Verfügbarkeitskomponenten beschrieben:

- [Sun Cluster 3.1 9/04 und Sun Cluster Agents](#)
- [High Availability Session Store 2005Q1](#)

## Sun Cluster 3.1 9/04 und Sun Cluster Agents

Die Sun Cluster-Software bietet Hochverfügbarkeits- und Skalierbarkeitsdienste für das Java Enterprise System sowie für Anwendungen, die auf der Java ES-Infrastruktur basieren.

Ein Cluster besteht aus einer Reihe von lose miteinander verknüpfter Computern (Clusterknoten), die gemeinsam eine einzige Client-Ansicht für Dienste, Systemressourcen und Daten bereitstellen. Intern nutzt der Cluster redundante Computer, Interconnects, Datenspeicher und Netzwerkschnittstellen, um clusterbasierte Dienste und Daten mit Hochverfügbarkeit zu versorgen. Die Sun Cluster-Software überwacht kontinuierlich den Zustand der Mitgliedsknoten und anderer Cluster-Ressourcen, wobei die interne Redundanz genutzt wird, um im Fehlerfall einen nahezu kontinuierlichen Zugriff auf diese Ressourcen zu gewährleisten.

Im Java ES Installer sind der Sun Cluster Core und die Sun Cluster-Agenten als einzeln installierbare Komponenten enthalten. Weitere Sun Cluster-Agenten sind auf separaten CDs erhältlich.

## High Availability Session Store 2005Q1

Sun Java System High Availability Session Store (HADB) bietet einen Datenspeicher, der genutzt werden kann, um Anwendungsdaten auch bei einem Ausfall verfügbar zu machen. Diese Möglichkeit ist besonders wichtig, um die einer Client-Sitzung zugeordneten Statusinformationen wiederherzustellen. Wenn während einer Sitzung ein Fehler auftritt müssen sonst in der neu erstellten Sitzung alle Vorgänge noch einmal wiederholt werden.

Die folgenden Java ES-Komponenten bieten Dienste, die Sitzungsstatusinformationen speichern: Application Server, Access Manager und Message Queue. Application Server ist jedoch die einzige dieser Komponenten, die HADB-Dienste nutzen kann, um während eines Ausfalls den Sitzungsstatus zu erhalten.

Das Java ES-Installationsprogramm stellt HADB als installierbare Einzelkomponente zur Verfügung. Für die HADB-Dienste werden jedoch sowohl eine Server- als auch eine Client-Unterkomponente benötigt.

## Beschreibung der Zugriffskomponenten

Zugriffskomponenten bieten Front-End-Zugriff auf Systemdienste, der häufig über Internetstandorte erfolgt, die sich außerhalb der Firewall des Unternehmens befinden. Die folgenden Java ES-Zugriffskomponenten werden in diesem Abschnitt beschrieben:

- [Sun Java System Communications Express 2005Q1](#)
- [Sun Java System Connector für Microsoft Outlook 6 2005Q1](#)
- [Sun Java System Directory Proxy Server 5 2005Q1](#)
- [Sun Java System Portal Server Secure Remote Access 6 2005Q1](#)

### Sun Java System Communications Express 2005Q1

Sun Java System Communications Express (Communications Express) sorgt für einen integrierten, webbasierten Kommunikations- und Zusammenarbeits-Client, der Endbenutzern eine Webschnittstelle zu Kalender-, E-Mail und Adressbuchdiensten bietet. Communications Express besteht aus drei Client-Modulen: Calendar, Address Book und Mail. Communications Express kann für den Zugriff auf Messaging Server oder Calendar Server oder auf beide konfiguriert werden und ist mit Sun Java System LDAP Schema, Version 1 (Schema 1) oder Schema 2 kompatibel.

Das Java ES-Installationsprogramm stellt Communications Express als installierbare Einzelkomponente zur Verfügung.

### Sun Java System Connector für Microsoft Outlook 6 2005Q1

Sun Java System Connector für Microsoft Outlook ermöglicht die Verwendung von Outlook als Desktop-Client für Sun Java Enterprise System. Der Connector ist ein Outlook-Plugin, das auf dem Desktop des Benutzers installiert werden muss.

Der Connector für Microsoft Outlook fragt Messaging Server nach Ordnerhierarchien und E-Mail-Nachrichten ab und konvertiert dann die Informationen in Messaging API-Eigenschaften (MAPI-Eigenschaften), die Outlook anzeigen kann. In ähnlicher Weise nutzt Connector WCAP zur Abfrage von Calendar Server nach Ereignissen und Aufgaben, die anschließend in MAPI-Eigenschaften umgewandelt werden. Anhand dieses Modells erstellt der Sun Java System Connector für Microsoft Outlook eine Outlook-Ansicht für den Endbenutzer aus zwei verschiedenen Informationsquellen: Mail aus Messaging Server und Kalenderinformationen aus Calendar Server.

Sun Java System Connector für Microsoft Outlook ist auf der Zubehör-CD enthalten (einschließlich Installationsprogramm).

## Sun Java System Directory Proxy Server 5 2005Q1

Sun Java System Directory Proxy Server (Directory Proxy Server) ist eine essentielle Komponente für alle geschäftskritischen Verzeichnisdienste für E-Commerce-Lösungen. Directory Proxy Server ist ein Protokoll-Gateway der LDAP-Anwendungsebene, das durch die Nutzung von Lastausgleich und Failover auf Anwendungsebene eine erweiterte Verzeichniszugriffssteuerung, Schemakompatibilität und Hochverfügbarkeit bietet.

Das Java ES-Installationsprogramm stellt Directory Proxy Server als installierbare Einzelkomponente zur Verfügung.

## Sun Java System Portal Server Secure Remote Access 6 2005Q1

Sun Java System Portal Server Secure Remote Access (Portal Server Secure Remote Access) erweitert Portal Server durch browserbasierten, sicheren Remote-Zugriff auf Portal Server-Inhalte und -Dienste, die über jeden Remote-Browser erfolgen können und die Client-Software überflüssig machen. Durch die Integration in Portal Server wird gewährleistet, dass die Benutzer auf sichere Weise auf den Inhalt und die Dienste zugreifen können, für die sie zugriffsberechtigt sind.

Das Java ES-Installationsprogramm stellt Portal Server Secure Remote Access als installierbare Einzelkomponente zur Verfügung. Bei Bedarf können die folgenden Portal Server Secure Remote Access-Unterkomponenten separat installiert werden:

- **Portal Server Secure Remote Access Core.**
- **Gateway:** Liefert eine Schnittstelle und Sicherheitsbarriere für das Intranet Ihres Unternehmens, das den Fernzugriff von außerhalb des Intranets ermöglicht. Gateway präsentiert einem Remote-Benutzer über eine einzige Schnittstelle Content sicher auf internen Web- und Anwendungsservern.
- **Netlet Proxy:** Ermöglicht dem Benutzer die sichere Ausführung gängiger TCP/IP-Dienste über das Internet und andere nicht sichere Netzwerke. Mit Netlet können Sie Anwendungen wie Telnet, SMTP, HTTP und Anwendungen mit festem Anschluss ausführen.
- **Rewriter Proxy:** Ermöglicht den sicheren Zugriff auf Webseiten von Unternehmens-Intranets von außerhalb des Intranets durch Umwandlung von Weblinks und das Erstellen von Regelsätzen für den Umgang mit Intranet-Webseiten.

## Beschreibung der Verwaltungskomponenten

Verwaltungskomponenten bieten Verwaltungsfunktionen für Systemdienste, wie Konfiguration und Überwachung. Die folgenden Java ES-Verwaltungskomponenten werden in diesem Abschnitt beschrieben:

- [Sun Java System Administration Server \(und Console\) 5 2005Q1](#)
- [Sun Java System Directory Preparation Tool 2005Q1](#)
- [Sun Java System Delegated Administrator 6 2005Q1](#)
- [Sun Remote Services Net Connect 2005Q1](#)

### Sun Java System Administration Server (und Console) 5 2005Q1

Sun Java System Administration Server und Server Console bieten zusammen ein grafisches Tool, mit dem Sie Directory Server, Messaging Server und Directory Proxy Server verwalten können. Der Administration Server verarbeitet Anforderungen für Server, die in einer Servergruppe unter demselben Root-Verzeichnis installiert sind, und startet dann die Programme, die zur Erfüllung der Anforderungen benötigt werden.

Server Console ist eine eigenständige Java-Anwendung, die in Kombination mit einer Instanz von Directory Server und einer Instanz von Administration Server in Ihrem Netzwerk eingesetzt wird. Server Console fungiert als Front-End-Verwaltungsanwendung für die Java ES-Software in Ihrem Unternehmen.

Das Java ES-Installationsprogramm stellt Server Console und Administration Server gemeinsam als einzeln installierbare Komponente zur Verfügung.

### Sun Java System Directory Preparation Tool 2005Q1

Das Sun Java System Directory Preparation Script wird für die Konfiguration von Directory Server mit dem Schema verwendet, das benötigt wird, damit Benutzer auf Messaging Server und Calendar Server zugreifen können.

### Sun Java System Delegated Administrator 6 2005Q1

Der Sun Java System Delegated Administrator ist ein Befehlszeilendienstprogramm (`commadmin`) für die Einrichtung von Benutzern, Gruppen, Domänen und Ressourcen für Calendar Server, Messaging Server und andere Java ES-Dienstleister.

Delegated Administrator wird automatisch zusammen mit dem Access Manager installiert.

## Sun Remote Services Net Connect 2005Q1

Sun Remote Services Net Connect ist eine Sammlung von Systemverwaltungsdiensten, die Ihnen helfen, Ihre IT-Umgebung besser zu steuern. Diese über das Web bereitgestellten Dienste geben Ihnen die Möglichkeit, Systeme zu überwachen, Leistungs- und Trendberichte zu erstellen und automatische Benachrichtigungen über Systemereignisse zu erhalten, sodass Sie schneller reagieren und potenzielle Schwierigkeiten in den Griff bekommen können, bevor sie zu Problemen werden.

Das Java ES-Installationsprogramm stellt Sun Remote Services Net Connect als installierbare Einzelkomponente zur Verfügung.

# Gemeinsam genutzte Komponenten

Gemeinsam genutzte Komponenten bieten die lokale Unterstützung für Dienste und Technologien, von denen die Systemdienstkomponenten und Dienstqualitätskomponenten von Java ES abhängen. Das Java ES-Installationsprogramm installiert automatisch alle gemeinsam genutzten Komponenten, die für die Unterstützung der anderen Java ES-Komponenten erforderlich sind, die auf einem Hostcomputer installiert sind.

Java Enterprise System enthält die unten aufgelisteten gemeinsam genutzten Komponenten:

- Ant (auf Jakarta ANT Java/XML basierendes Tool)
- Apache SOAP (Simple Object Access Protocol) Runtime
- Berkeley DB
- Common Agent Container
- ICU (International Components for Unicode)
- J2SE™ (Java 2 Platform, Standard Edition) Platform 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaHelp™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXM (Java API for XML Messaging) Client Runtime

- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java-API für XML-basierten Remote Procedure Call) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- Perl LDAP, einschließlich NSPERL
- SAAJ (SOAP mit Anhangs-API für Java)
- SAML (Security Assertions Markup Language)
- SASL (Simple Authentication and Security Layer)
- SNMP (Simple Network Management Protocol) Peer
- Sun Explorer Data Collector (nur Solaris)
- Sun Java Monitoring Framework
- Sun Java Web Console
- Tomcat Servlet JSP Container
- XML C Library (libxml)
- WSCL (Web Services Common Library)

Gemeinsam genutzte Komponenten

## A

- Abhängigkeiten [42, 61](#)
- Abhängigkeitsprüfung, Installationsprogramm [61](#)
- Access Manager
  - als Infrastrukturdienst [41](#)
  - als Systemdienstkomponente [22](#)
  - Beschreibung von [82](#)
- Administration Server
  - als Dienstqualitätskomponente [25](#)
  - Beschreibung von [91](#)
- Ant (auf Jakarta ANT Java/XML basierendes Tool) [92](#)
- Anwendungen
  - Unternehmen, *Siehe* Verteilte Unternehmensanwendungen
  - verteilte, *Siehe* Verteilte Unternehmensanwendungen
- Anwendungsdienste [19, 38](#)
- Anwendungsfälle
  - Definition [80](#)
  - vorgestellt [71](#)
- Anwendungskomponenten
  - Definition [57](#)
  - in logischen Schichtenarchitekturen [44](#)
- Apache SOAP Runtime [92](#)
- Application Server
  - als Infrastrukturdienst [41](#)
  - als Systemdienstkomponente [22](#)
  - Beschreibung von [83](#)
- Architektur
  - Bereitstellung [73](#)
  - Definition [57](#)

- Dimensionen, *Siehe* Architekturdimensionen
  - logisch [71](#)
  - Lösung [36](#)
  - Referenz [74](#)
  - vorgestellt [35](#)
- Architekturdimensionen
  - Dienstqualität [47](#)
  - Infrastrukturdienstabhängigkeiten [38](#)
  - logische Schichten [43](#)
  - Synthese [52](#)
- Aufgaben, Java Enterprise System [27, 69](#)
- Aufrüstungen, *Siehe* Einführungsszenarien
- Authentifizierung [65](#)
- Autorisierung [66](#)

## B

- Benutzer, *Siehe* Endbenutzer
- Benutzereinrichtung [75](#)
- Benutzereintrag [63](#)
- Benutzerkategorien
  - Architekturdesigner [29](#)
  - Außendiensttechniker [29](#)
  - Bevollmächtigter Administrator [29](#)
  - Geschäftsplaner [29](#)
  - IT-Manager [29](#)
  - spezialisierte Systemadministrator [29](#)
  - Systemadministrator [29](#)
  - Systemanalyst [29](#)
  - Systemintegrator [29](#)

- Benutzerprofile 29
- Benutzerzusammenarbeitsdienste 40
- Bereitstellung
  - Architektur 73
  - Definition 79
  - Entwicklung und Anpassung 77
  - Implementierung 75
  - Konzept 73
  - Lebenszyklusabschnitt 72
  - Phasen des Lebenszyklus 72
  - Produktionseinsatz 78
  - Prototyp-Tests 77
  - Referenzarchitekturen 74
  - Szenarios, *Siehe* Bereitstellungsszenarios
- Bereitstellungsarchitekturen
  - Beziehung zu geschichteten Architekturen 46
  - Definition 57
  - Entwerfen von 73
  - vorgestellt 35
- Bereitstellungsnachbereitung
  - Definition 80
  - Phasen des Lebenszyklus 78
- Bereitstellungsszenarios
  - Definition 79
  - vorgestellt 71
- Bereitstellungsvorbereitung
  - Definition 79
  - Phasen des Lebenszyklus 71
- Berkeley DB 92
- Betriebssystemdienste 39

## C

- Calendar Server
  - als Infrastrukturdienst 41
  - als Systemdienstkomponente 22
  - Beschreibung 84
- CAPI (Java Calendar API) 93
- Clients
  - Definition 57
  - Systemdienstkomponenten und 22
- Cluster, *Siehe* Sun Cluster
- Common Agent Container 92

- Communications Express
  - als Dienstqualitätskomponente 25
  - Beschreibung von 89
- Connector für Microsoft Outlook
  - als Dienstqualitätskomponente 25
  - Beschreibung von 89

## D

- Deinstallationsprogramm 62
- Delegated Administrator
  - als Dienstqualitätskomponente 25
  - Beschreibung von 91
- Dienste
  - Definition 33
  - Geschäft 44
  - Hochverfügbarkeit 50, 88
  - Infrastruktur 18
  - Infrastruktur, *Siehe* Verteilte Infrastrukturdienste
  - Skalierbarkeit 50, 88
  - Web 45
- Dienstqualitätsanforderungen
  - Latente Kapazität 48
  - Leistung 48
  - Sicherheit 48, 49, 50
  - Skalierbarkeit 48, 49, 50
  - Verfügbarkeit 48, 50
  - Zweckmäßigkeit 48, 50
- Dienstqualitätskomponenten
  - Beschreibung von 87
  - Definition 33
  - vorgestellt 23
- Directory Preparation Tool
  - als Dienstqualitätskomponente 25
  - Beschreibung von 91
- Directory Proxy Server
  - als Dienstqualitätskomponente 24
  - als Systemkomponente 49
  - Beschreibung von 90
- Directory Server
  - als Infrastrukturdienst 41
  - als Systemdienstkomponente 22
  - Beschreibung von 84
- Dokumentation, Java ES 12

**E**

- Einführungsszenarien, Java ES
  - Aufrüstung 30
  - Definition 32
  - Erweiterung 30
  - Info 30
  - Neues System 30
  - Verbesserung 30
- Einrichten von Benutzern 75
- Einzelidentität
  - Definition 68
  - vorgestellt 63
- EJB-Komponenten 45
- Endbenutzer
  - Definition 33
  - Verteilte Unternehmensanwendungen und 18
- Entwicklung
  - als Bereitstellungsaufgabe 77
  - Definition 79
- Ermitteln installierter Software 61
- Erweiterungen, *Siehe* Einführungsszenarien

**G**

- Gemeinsam genutzte Komponenten 92
  - Definition 33
  - vorgestellt 26
- Geschäftsdienste
  - Definition 57
  - Präsentationsschicht und 44

**H**

- Hardware, Java ES-Einführungsszenarien und 31
- High Availability Session Store
  - als Dienstqualitätskomponente 24
  - Beschreibung von 88

**I**

- ICU (International Components for Unicode) 92
- Identität
  - Dienste 19, 62
  - Einzelner Benutzer 63
  - Verwaltung 62
- Implementierungsspezifikationen 74, 75
- Infrastruktur
  - Dienstabhängigkeiten, *Siehe* Verteilte Dienste für verteilte Unternehmensanwendungen 18
- Instant Messaging
  - als Infrastrukturdienst 41
  - als Systemdienstkomponente 22
  - Beschreibung von 84
- Integration
  - Dienste 40
  - Funktionen, *Siehe* Integrationsfunktionen
  - Java ES-Einführungsszenarien und 31
- Integrationsfunktionen
  - Gemeinsam genutzte Komponenten 21
  - Identität und Sicherheit 21, 62
  - integriertes Installationsprogramm 21, 60
  - vorgestellt 21

**J**

- J2EE
  - Komponenten 44
  - Plattform 23
  - verteiltes Komponentenmodell 45
- J2ME-Plattform 44
- J2SE (Java 2 Platform, Standard Edition) 26, 92
- JAF (JavaBeans™ Activation Framework) 92
- JATO (Java Studio Enterprise Web Application Framework) 92
- Java Servlet-Komponenten 44
- JavaHelp 92
- JavaMail 92
- JAXB (Java Architecture for XML Binding) 92

JAXM (Java API for XML Messaging) 92  
JAXP (Java API for XML Processing) 93  
JAXR (Java API for XML Registries) 93  
JAX-RPC 93  
JDMK (Java Dynamic Management™ Kit) 93  
JMS (Java Message Service) 23  
JSP-Komponenten 44  
JSS (Java Security Services) 26, 93

## K

Kommunikationsdienste 19  
Komponenten 18  
  Abhängigkeiten 42  
  Definition 32  
  Dienstqualität 23, 87  
  EJB 45  
  Ermitteln installierter Versionen von 61  
  gemeinsam genutzt 26, 92  
  J2EE 44  
  JSP 44  
  Servlet 44  
  System, *Siehe* Systemkomponenten  
  Systemdienst 22, 82  
  und Infrastrukturdienste 41  
  Verfügbarkeit 24  
  verteilt 18  
  Verwaltung 25  
  Zugriff 24  
KT Search Engine (KTSE) 93

## L

Latente Kapazitätsanforderungen 48  
Laufzeitdienste 39  
LDAP 45, 68  
LDAP C SDK 93  
LDAP Java SDK 93

Lebenszyklusphasen  
  Bereitstellung 27, 72  
  Bereitstellungsnachbereitung 27, 78  
  Bereitstellungsvorbereitung 27, 71  
Leistungsanforderungen 48  
Linux 41  
Logische Architekturen  
  Beispiel 54  
  Definition 57  
  Info 71  
  Infrastrukturdienstebenen und 38  
  vorgestellt 35  
logische Client-Schicht 44  
Lösungen, Java ES  
  Architektur 35  
  Beispiel 53  
  kundenspezifisch und Out-of-the-Box 53  
  Lebenszyklus 27

## M

Message Queue  
  als Infrastrukturdienst 41  
  als Systemdienstkomponente 23  
  Beschreibung von 85  
Messaging Server  
  als Infrastrukturdienst 41  
  als Systemdienstkomponente 23  
  Beschreibung von 86  
Messaging-Dienste 39  
Middleware-Dienste 38  
Migration, Java ES-Einführungsszenarien und 31

## N

Netzwerktransportdienste 39  
NSPR (Netscape Portable Runtime) 26, 93  
NSS (Network Security Services) 26, 93

**P**

- Perl LDAP 93
- Persistenzdienste 39
- Plattformdienste 38
- Portal Server
  - als Infrastrukturdienst 41
  - als Systemdienstkomponente 23
  - Beschreibung von 86
- Portal Server Secure Remote Access
  - als Dienstqualitätskomponente 24
  - als Systemkomponente 50
  - Beschreibung von 90
- Produktionseinsatz 78
- Prototyp-Erstellung 77

**R**

- Referenz-Bereitstellungsarchitekturen
  - Definition 80
  - vorgestellt 74
- Remote Services Net Connect
  - als Dienstqualitätskomponente 25
  - Beschreibung von 92
- Richtlinien
  - Autorisierung 66
  - Definition 68

**S**

- SAAJ (SOAP mit Anhangs-API für Java) 93
- SAML (Security Assertions Markup Language) 93
- SASL (Simple Authentication and Security Layer) 93
- Schichten, logische
  - Anwendungsarchitektur und 43
  - Client 44
  - Daten 45
  - Geschäftsdienst 45
  - Präsentation 44
- Schulung, Java ES-Einführungsszenarien und 31

- Server
  - Definition 58
  - eigenständig 45
- Sicherheit
  - Anforderungen 48, 49, 50
  - Dienste 19
  - Richtliniendienste 40
- Single Sign-On
  - Definition 68
  - Implementierung von 66
  - Infrastrukturdienstebenen und 40
  - Java ES-Funktion 22, 64
- Skalierbarkeit
  - Anforderungen 48, 49, 50
  - Dienste 50, 88
- SNMP (Simple Network Management Protocol) 93
- SOAP (Apache) 92
- Solaris 41
- Sprachunterstützung 60
- Sun Cluster
  - Agenten 51
  - als Dienstqualitätskomponente 24
  - als Verfügbarkeitsdienst 50
  - Beschreibung von 88
- Sun Explorer Data Collector 93
- Sun Java Monitoring Framework 93
- Sun Java System-Produkte
  - Access Manager, *Siehe* Access Manager
  - Administration Server, *Siehe* Administration Server
  - Application Server, *Siehe* Application Server
  - Calendar Server, *Siehe* Calendar Server
  - Communications Express, *Siehe* Communications Express
  - Connector für Microsoft Outlook, *Siehe* Connector für Microsoft Outlook
  - Delegated Administrator, *Siehe* Delegated Administrator
  - Directory Preparation Tool, *Siehe* Directory Preparation Tool
  - Directory Proxy Server, *Siehe* Directory Proxy Server
  - Directory Server, *Siehe* Directory Server
  - High Availability Session Store, *Siehe* High Availability Session Store

- Instant Messaging, *Siehe* Instant Messaging
- Message Queue, *Siehe* Message Queue
- Messaging Server, *Siehe* Messaging Server
- Portal Server Secure Remote Access, *Siehe* Portal Server Secure Remote Access
- Portal Server, *Siehe* Portal Server
- Web Server, *Siehe* Web Server
- Sun Java Web Console 93
- System
  - Dienste 18
  - Komponenten, *Siehe* Systemkomponenten
  - Konfiguration 62
- Systemdienste
  - Definition 33
  - Info 22
- Systemdienstkomponenten
  - Abhängigkeiten 42
  - Definition 33
  - vorgestellt 22
- Systemkomponenten
  - Definition 33
  - Dienstqualitätskomponenten 23
  - Gemeinsam genutzte Komponenten 26, 92
  - Info 21
  - Systemdienst 82
  - Systemdienstkomponenten, *Siehe* Systemdienstkomponenten

## T

- Tomcat Servlet JSP Container 93

## V

- Verbesserungen, *Siehe* Einführungsszenarien
- Verfügbarkeit
  - Anforderungen 48, 50
  - Dienste 50, 88
- Verfügbarkeitsdienste 20

- Verfügbarkeitskomponenten
  - Beschreibung von 87
  - vorgestellt 24
- Verteilte
  - Anwendungen, *Siehe* Verteilte Unternehmensanwendungen
  - Dienste, *Siehe* Verteilte Dienste
- verteilte Dienste
  - Anwendungsebene 38
  - Benutzerzusammenarbeit 40
  - Identität 19
  - Infrastruktur 18
  - Integration 40
  - Kommunikation und Zusammenarbeit 19
  - Laufzeit 19, 39
  - Messaging 39
  - Middleware 38
  - Netzwerktransport 39
  - Persistenz 39
  - Plattform 38, 39
  - Sicherheit 19, 40
  - Überblick 18
  - Verfügbarkeit 20
  - Verwaltung 20
  - Web 19
  - Zugang 19
  - Zugriff 20
- Verteilte Unternehmensanwendungen
  - Definition 32
  - Info 18
  - Infrastruktur für 18
- Verwaltungsdienste 20
- Verwaltungskomponenten
  - Beschreibung von 91
  - vorgestellt 25
- Verzeichnisse
  - als Benutzerdatenspeicher 63
  - Definition 68

**W**

- Web Server
  - als Infrastrukturdienst [41](#)
  - als Systemdienstkomponente [23](#)
  - Beschreibung von [87](#)
- Webdienste [19](#)
  - Definition [58](#)
  - J2EE-Komponenten und [45](#)
- WSCL (Web Services Common Library) [93](#)

**X**

- XML C Library [93](#)

**Z**

- Zugangsdienste [19](#)
- Zugriffsdienste [20](#)
- Zugriffskomponenten
  - Beschreibung von [89](#)
  - vorgestellt [24](#)
- Zusammenarbeitsdienste [19](#)
- Zweckmäßigkeitserfordernisse [48, 50](#)

