



# Sun Java Enterprise System 2005Q4 Technischer Überblick

---

Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
U.S.A.

Teilenr.: 819-3588  
Oktober 2005

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Alle Rechte vorbehalten.

Dieses Produkt oder Dokument ist urheberrechtlich und durch internationale Verträge geschützt. Dieses Produkt wird unter Lizenzen vertrieben, die die Verwendung, das Kopieren, die Verbreitung und Dekompilierung des Produkts verbieten. Kein Teil dieses Produkts oder Dokuments darf ohne die vorherige schriftliche Genehmigung von Sun und seinen Lizenzgebern (falls vorhanden) in keiner Weise und mit keinerlei Mitteln vervielfältigt werden. Software von Fremdanbietern, einschließlich Schriftartentechnologien, ist urheberrechtlich geschützt und lizenziert durch Sun-Lieferanten.

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 in anderen Ländern und exklusiv durch X/Open Company, Ltd. lizenziert.

Sun, Sun Microsystems, das Sun-Logo, docs.sun.com, AnswerBook, AnswerBook2, und Solaris 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 das SPARC-Warenzeichen aufweisen, basieren auf der von Sun Microsystems, Inc. entwickelten Architektur.

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.

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

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

---

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPOUDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



051220@13215



# Inhalt

---

<b>Vorwort</b>	<b>13</b>
<b>1 Einführung in Java Enterprise System</b>	<b>19</b>
Wozu benötigen Sie Java Enterprise System?	19
Java Enterprise System-Komponenten	21
Systemdienstkomponenten	22
Dienstqualitätskomponenten	24
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>2 Lösungsarchitekturen von Java Enterprise System</b>	<b>35</b>
Java Enterprise System-Architekturframework	35
Dimension 1: Infrastrukturdienstabhängigkeiten	37
Dimension 2: Logische Schichten	42
Dimension 3: Dienstqualität	46
Synthese der drei Architekturdimensionen	50
Beispiel für Java Enterprise System-Lösungsarchitektur	50
Szenario der Unternehmenskommunikation	51
Logische Architektur des Beispielszenarios	52
Bereitstellungsarchitektur des Beispielszenarios	53
In diesem Kapitel enthaltene Schlüsselbegriffe	53

<b>3</b>	<b>Integrationsfunktionen von Java Enterprise System</b>	<b>55</b>
	Das integrierte Installationsprogramm von Java Enterprise System	55
	Überprüfung bereits vorhandener Software	56
	Überprüfung der Abhängigkeiten	56
	Erstkonfiguration	57
	Deinstallation	57
	Integrierte Identitäts- und Sicherheitsdienste	58
	Einzelidentität	58
	Authentifizierung und Single Sign-On	59
	Autorisierung	61
	In diesem Kapitel enthaltene Schlüsselbegriffe	62
<b>4</b>	<b>Aufgaben des Lösungslebenszyklus von Java Enterprise System</b>	<b>63</b>
	Aufgaben des Lösungslebenszyklus	63
	Bereitstellungsvorbereitung	65
	Bereitstellung	66
	Bereitstellungskonzept	67
	Bereitstellungsimplementierung	69
	Bereitstellungsnachbereitung	72
	In diesem Kapitel enthaltene Schlüsselbegriffe	72
<b>5</b>	<b>Referenzliste: Java Enterprise System-Komponenten</b>	<b>75</b>
	Beschreibungen der Systemdienstkomponenten	76
	Sun Java System Access Manager 7 2005Q4	76
	Sun Java System Application Server Enterprise Edition 8.1 2005Q4	77
	Sun Java System Calendar Server 6 2005Q4	77
	Sun Java System Directory Server 5 2005Q4	78
	Sun Java System Instant Messaging 7 2005Q4	78
	Sun Java System Message Queue 3.6 2005Q4	78
	Sun Java System Messaging Server 6 2005Q4	79
	Sun Java System Portal Server 6 2005Q4	80
	Sun Java System Service Registry 3 2005Q4	80
	Sun Java System Web Server 6.1 2005Q4	80
	Beschreibung der Dienstqualitätskomponenten	81
	Beschreibung der Verfügbarkeitskomponenten	81
	Beschreibung der Zugriffskomponenten	82
	Beschreibung der Verwaltungskomponenten	84

Gemeinsam genutzte Komponenten 85

**Index** 87



# Tabellen

---

<b>TABELLE 1-1</b>	Java ES-Systemdienstkomponenten	23
<b>TABELLE 1-2</b>	Java ES-Verfügbarkeitskomponenten	25
<b>TABELLE 1-3</b>	Java ES-Zugriffskomponenten	25
<b>TABELLE 1-4</b>	Java ES-Verwaltungskomponenten	26
<b>TABELLE 1-5</b>	Java ES-Benutzerkategorien für Aufgaben im Lebenszyklus	29
<b>TABELLE 1-6</b>	Aspekte unterschiedlicher Java ES-Einführungsszenarien	31
<b>TABELLE 2-1</b>	Beziehungen zwischen Java ES-Systemdienstkomponenten	41
<b>TABELLE 2-2</b>	Dienstqualitäten mit Auswirkung auf die Lösungsarchitektur	46
<b>TABELLE 2-3</b>	Dienstqualitätskomponenten und die beeinflussten Systemqualitäten	48
<b>TABELLE 2-4</b>	Zusammenfassung der Geschäftsanforderungen: Kommunikationsszenario	51





# Abbildungen

---

<b>ABBILDUNG 1-1</b>	Erforderliche Unterstützung für verteilte Unternehmensanwendungen	20
<b>ABBILDUNG 1-2</b>	Kategorien der Java ES-Komponenten	22
<b>ABBILDUNG 1-3</b>	Phasen des Lebenszyklus einer Lösung und Benutzerkategorien	28
<b>ABBILDUNG 2-1</b>	Dimensionen der Java ES-Lösungsarchitektur	36
<b>ABBILDUNG 2-2</b>	Dimension 1: Infrastrukturdienstebenen	38
<b>ABBILDUNG 2-3</b>	Java ES-Systemdienstkomponenten	40
<b>ABBILDUNG 2-4</b>	Dimension 2: Logische Schichten für verteilte Unternehmensanwendungen	43
<b>ABBILDUNG 2-5</b>	Messaging Server: Beispiel für eine Schichtenarchitektur	45
<b>ABBILDUNG 2-6</b>	Verfügbarkeitskonzept mit Sun Cluster-Knoten	49
<b>ABBILDUNG 2-7</b>	Logische Architektur des Unternehmenskommunikationsszenarios	52
<b>ABBILDUNG 3-1</b>	Ein einzelner Benutzereintrag unterstützt viele Dienste	59
<b>ABBILDUNG 3-2</b>	Authentifizierungssequenz	60
<b>ABBILDUNG 3-3</b>	Autorisierungssequenz	61
<b>ABBILDUNG 4-1</b>	Aufgaben des Lösungslebenszyklus	64
<b>ABBILDUNG 4-2</b>	Spezifizieren eines Bereitstellungsszenarios	66
<b>ABBILDUNG 4-3</b>	Umsetzen eines Bereitstellungsszenarios in eine Bereitstellungsarchitektur	67



# Beispiele

---



# Vorwort

---

Das Handbuch *Java Enterprise System Technischer Überblick* bietet eine Einführung in die technischen und konzeptionellen Grundlagen von Java Enterprise System. 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-Dokumentationsatz 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.

---

## An wen richtet sich dieses Handbuch

Das Handbuch *Java Enterprise System Technischer Überblick* richtet sich an Personen, die für den Entwurf, die Bereitstellung oder Wartung von Software-Lösungen verantwortlich sind, die auf Java Enterprise System basieren. Zu dieser großen Zielgruppe gehören beispielsweise Geschäftsanalysten, Systemarchitekturdesigner, Außendiensttechniker und Systemadministratoren.

Die Leser des Handbuchs *Java Enterprise System Technischer Überblick* sollten mit folgenden Technologien vertraut sein:

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

---

# Java Enterprise System-Dokumentationssatz

Der Java ES-Dokumentationssatz beschreibt die Bereitstellungsplanung und die Systeminstallation. Der URL für die Systemdokumentation lautet <http://docs.sun.com/coll/1286.1>. Eine Einführung in Java ES finden Sie in den in der folgenden Tabelle aufgeführten Handbüchern. Lesen Sie die Handbücher in der hier gezeigten Reihenfolge.

**TABELLE P-1** Java Enterprise System-Dokumentation

Dokument	Inhalt
<i>Sun Java Enterprise System 2005Q4 Versionshinweise</i>	Enthält die neuesten Informationen über Java ES, einschließlich bekannter Probleme. Zudem verfügen die Komponenten über ihre eigenen Versionshinweise.
<i>Übersicht über die Dokumentation zu Sun Java Enterprise System 2005Q4</i>	Enthält Beschreibungen der Dokumentation zu Java ES als System als auch für die einzelnen Komponenten.
<i>Sun Java Enterprise System 2005Q4 Technischer Überblick</i>	Bietet eine Einführung in die technischen und konzeptionellen Grundlagen von Java ES. Beschreibt die Komponenten, die Architektur sowie Prozesse und Funktionen.
<i>Sun Java Enterprise System 2005Q4 Handbuch zur Bereitstellungsplanung</i>	Bietet eine Einführung in die Planung und die Konzeptentwicklung von Bereitstellungslösungen für Unternehmen, die auf Java ES basieren. Beschreibt die Grundlagen und Prinzipien der Bereitstellungsplanung und des Bereitstellungskonzepts sowie den Lebenszyklus einer Lösung und bietet allgemeine Beispiele und Strategien, die Sie bei der Planung der unternehmensweiten Bereitstellung von Lösungen mit Java ES anwenden können.
<i>Java ES System 2005Q4 Handbuch zur Installationsplanung</i>	Unterstützt Sie bei der Entwicklung der Implementierungsspezifikationen für die Hardware, das Betriebssystem und die Netzwerkaspekte Ihrer Java ES-Bereitstellung. Beschreibt Probleme, wie beispielsweise Komponentenabhängigkeiten, die in Ihrem Installations- und Konfigurationsplan berücksichtigt werden müssen.

**TABELLE P-1** Java Enterprise System-Dokumentation (Fortsetzung)

Dokument	Inhalt
<i>Sun Java Enterprise System 2005Q4 Installationshandbuch für UNIX</i>	Führt Sie durch die Installation von Java ES für das Solaris- oder das Linux-Betriebssystem. Es wird außerdem beschrieben, wie Sie die Komponenten nach der Installation konfigurieren und wie Sie überprüfen, ob die Komponenten ordnungsgemäß ausgeführt werden.
<i>Java ES System 2005Q4 Installation-Verweis</i>	Bietet zusätzliche Informationen zu Konfigurationsparametern, enthält Arbeitsblätter, die Sie für Ihren Konfigurationsplan verwenden können, sowie Referenzmaterialien, wie beispielsweise Standardverzeichnisse und standardmäßige Anschlussnummern.
<i>Sun Java Enterprise System 2005Q1 Deployment Example Series: Evaluation Scenario</i>	Beschreibt die Installation von Java ES auf einem System, das Herstellen einer Reihe wichtiger freigegebener Dienste und Netzwerkdienste und die Einrichtung der Benutzerkonten, mit denen auf die von Ihnen eingerichteten Dienste zugegriffen werden kann.
<i>Sun Java Enterprise System 2005Q4 Auf[rück]stellungshandbuch</i>	Enthält Anleitungen für das Aufrüsten von Java ES unter dem Solaris-Betriebssystem oder in der Linux-Betriebssystemumgebung.
<i>Sun Java Enterprise System Glossary</i>	Definiert die Begriffe, die in der Java ES-Dokumentation verwendet werden.

---

## Typografische Konventionen

In der nachfolgenden Tabelle werden die in diesem Handbuch verwendeten typografischen Änderungen erläutert.

**TABELLE P-2** Typografische Konventionen

Schriftart	Bedeutung	Beispiel
AaBbCc123	Befehlsnamen, Dateinamen, Verzeichnispfadnamen und Bildschirmcomputerausgaben	Bearbeiten Sie Ihre <code>.login</code> -Datei. Verwenden Sie <code>ls -a</code> zum Auflisten aller Dateien. <code>machine_name% you have mail.</code>

**TABELLE P-2** Typografische Konventionen (Fortsetzung)

Schriftart	Bedeutung	Beispiel
<b>AaBbCc123</b>	Eingabe durch den Benutzer (im Gegensatz zur Computerausgabe auf dem Bildschirm)	Computername% <b>su</b> Password:
<i>AaBbCc123</i>	Platzhalter, der durch tatsächliche Namen oder Werte ersetzt werden muss.	Der Befehl zum Entfernen einer Datei lautet <i>rm filename</i> .
<i>AaBbCc123</i>	Buchtitel, neue Begriffe, hervorzuhebende Begriffe (beachten Sie, dass einige der hervorzuhebenden Begriffe online fettgedruckt angezeigt werden)	Lesen Sie Kapitel 6 im <i>User's Guide</i> . Ein <i>Cache</i> ist eine lokal gespeicherte Kopie. Speichern Sie die Datei <i>nicht</i> .

---

## Online-Zugriff auf Sun-Ressourcen

Auf der Website docs.sun.com<sup>SM</sup> können Sie online auf die technische Dokumentation von Sun zugreifen. Sie können das docs.sun.com-Archiv durchsuchen oder nach einem bestimmten Buchtitel oder Thema suchen. Die Handbücher sind als Online-Dateien im PDF-Format (Portable Document Format) und HTML-Format (Hypertext Markup Language) verfügbar. Beide Formate können von unterstützenden Technologien für Benutzer mit Behinderungen gelesen werden.

Zugriff auf die folgenden Sun-Ressourcen erhalten Sie unter der Adresse <http://www.sun.com>:

- Downloads von Sun-Produkten
- Services und Lösungen
- Support (einschließlich Patches und Updates)
- Schulung
- Forschung
- Communities (z. B. Sun Developer Network)

---

## Verwandte Website-Referenzen von Drittanbietern

In der vorliegenden Dokumentation wird auf URLs von Drittanbietern verwiesen, über die zusätzliche relevante Informationen zur Verfügung gestellt werden.



---

**Hinweis** – Sun ist nicht haftbar für die Verfügbarkeit der Websites Dritter, die in diesem Dokument erwähnt werden. Sun unterstützt keine Inhalte, Werbung, Produkte oder sonstige Materialien, die auf oder über solche Websites oder Ressourcen verfügbar sind, und übernimmt keine Verantwortung oder Haftung dafür. Sun ist nicht verantwortlich oder haftbar für tatsächliche oder vermeintliche Schäden oder Verluste, die durch oder in Verbindung mit der Verwendung von über solche Websites oder Ressourcen verfügbaren Inhalten, Waren oder Dienstleistungen bzw. dem Vertrauen darauf entstanden sind oder angeblich entstanden sind.

---

---

## Kommentare sind willkommen

Sun möchte seine Dokumentation laufend verbessern. Ihre Kommentare und Vorschläge sind daher immer willkommen. Um Ihre Kommentare einzusenden, navigieren Sie zu <http://docs.sun.com> und klicken Sie auf "Send Comments". Geben Sie im Online-Formular den vollständigen Dokumenttitel und die Teilenummer an. Die Teilenummer ist eine 7- oder 9-stellige Nummer, die Sie auf der Titelseite des Handbuchs oder in dem URL des Dokuments finden. Die Teilenummer für dieses Handbuch lautet beispielsweise 819-3588.



# 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 19
- „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?

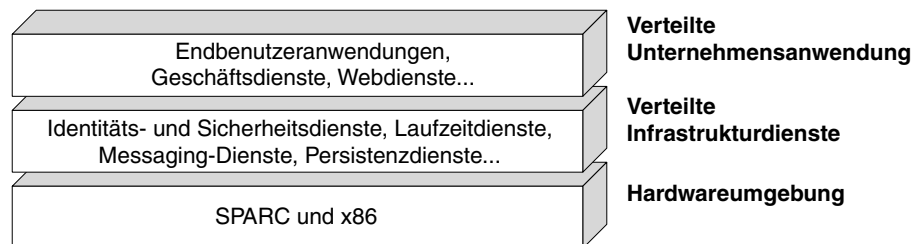
Die Geschäftsanforderungen von heute fordern Softwarelösungen, die über eine Netzwerk- oder Internetumgebung hinweg verteilt sind sowie hohe Leistung, Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit bieten.

Java Enterprise System bietet Infrastrukturdienste, die benötigt werden, um *distributed enterprise applications* zu unterstützen, das heißt, Anwendungen, die in der Regel folgende Merkmale aufweisen:

- **Verteilt.** Die Anwendung besteht aus interagierenden Software- **components**, die über eine vernetzte Umgebung verteilt werden, die geografisch entfernte Standorte umfassen kann. Diese verteilten Komponenten, die auf den verschiedenen Computern der Umgebung ausgeführt werden, arbeiten zusammen, um spezielle Geschäftsfunktionen für die **end users** und andere Geschäftsanwendungen bereitzustellen.
- **Unternehmensstärke.** Umfang und Größe der Anwendung erfüllen die Anforderungen einer Produktionsumgebung oder eines Internetdienstanbieters. 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 zugrunde liegende Infrastruktur von **services**, die den verteilten Komponenten die Kommunikation untereinander, die Koordination der Arbeit, die Implementierung eines sicheren Zugriffs usw. ermöglichen. Diese Infrastrukturdienste werden ihrerseits 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 sind. 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, Telearbeiter, 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 Zusammenarbeitdienste.** 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. Die Dienste basieren auf der Enterprise Edition (J2EE™)-Technologie für die Java 2-Plattform.

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 und -komponenten 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 gemeinsamer Bibliotheken synchronisiert.

- Alle Java ES-Komponenten werden mithilfe eines einzigen Installationsprogramms installiert.
- Alle Java ES-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 *system components* können in drei Hauptkategorien unterteilt werden, wie in der folgenden Darstellung gezeigt:

- **Systemdienstkomponenten.** Diese Komponenten bieten die grundlegenden Java Enterprise System-Infrastrukturdienste, die für die Unterstützung verteilter Unternehmensanwendungen benötigt werden.
- **Dienstqualitätskomponenten.** Diese Komponenten verbessern die Verfügbarkeit, Sicherheit, Skalierbarkeit, Zweckmäßigkeit sowie andere Bereiche der Systemdienstkomponenten und verteilten Anwendungskomponenten.
- **Gemeinsam genutzte Komponenten.** Diese Komponenten bilden die Umgebung, in der viele Systemdienst- und Dienstqualitätskomponenten ausgeführt werden.

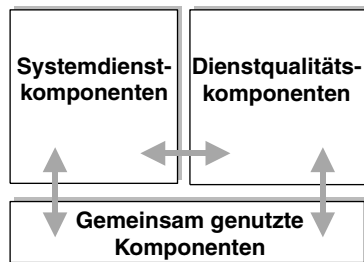


ABBILDUNG 1-2 Kategorien der Java ES-Komponenten

## Systemdienstkomponenten

Einige Java ES-Komponenten bilden die Hauptdienste für die Unterstützung verteilter Softwarelösungen. Zu diesen *system services* gehören Zugangsdienste, Kommunikations- und Zusammenarbeitsdienste, Identitäts- und Sicherheitsdienste, Webcontainerdienste sowie J2EE-Anwendungsdienste.

Die folgende Tabelle enthält eine kurze Beschreibung der *system service components*, die diese verteilten Dienste bereitstellen, und der von diesen Diensten angebotenen Dienste. Jede Systemdienstkomponente ist ein Servervorgang mit mehreren Threads, der eine Vielzahl von Clients unterstützen kann. Weitere Informationen zu den einzelnen Komponenten finden Sie unter „Beschreibungen der Systemdienstkomponenten“ auf Seite 76.

**TABELLE 1–1** Java ES-Systemdienstkomponenten

<b>Komponente</b>	<b>Bereitgestellte Systemdienste</b>
Sun Java System Access Manager	Stellt Dienste für die Zugriffsverwaltung und die digitale Identitätsverwaltung bereit. Zu den Zugriffsverwaltungsdiensten zählen 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 JEnterprise 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. Application Server stellt außerdem Webcontainterdienste bereit.
Sun Java System Calendar Server	Dieser Dienst bietet Kalender- und Planungsdienste für Endbenutzer und Endbenutzergruppen. Calendar Server umfasst 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.
Sun Java System Message Queue	Bietet zuverlässigen, asynchronen Messaging-Austausch 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 auf mehrere Nachrichtenspeicher gleichzeitig zugreifen und ermöglicht die Filterung von Inhalten zur Zurückweisung unerwünschter E-Mails und zur Vermeidung von Virenangriffen.

**TABELLE 1–1** Java ES-Systemdienstkomponenten (Fortsetzung)

Komponente	Bereitgestellte Systemdienste
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 stellt darüber hinaus eine konfigurierbare Suchmaschine bereit.
Sun Java System Web Server	Bietet J2EE™-Webcontainerdienste für Java-Webkomponenten, wie die Komponenten Java Servlet und JavaServer Pages™ (JSP™). Web Server unterstützt zudem andere Webanwendungstechnologien für die Bereitstellung von statischem und dynamischem Webinhalt wie CGI-Skripts und Java System Active Server Pages.
Sun Java System Service Registry	Bietet eine Registry und ein Repository für die Unterstützung von Service-Oriented Architecture-(SOA-)Anwendungen. Service Registry implementiert Industriestandards für die Registrierung und das Auffinden von Webdiensten sowie für die Verwaltung der dazugehörigen Informationen und Fakten, Artefakte (z. B. XMLSchema, Unternehmensprozessregeln, Zugriffssteuerung, Versionskontrolle).

## Dienstqualitätskomponenten

Neben den in [Tabelle 1–1](#) gezeigten Systemdienstkomponenten enthält Java Enterprise System eine Vielzahl von Komponenten, mit denen die Qualität der von den Systemdienstkomponenten bereitgestellten Dienste verbessert wird. Dienstqualitätskomponenten können auch kundenspezifisch entwickelte Anwendungsdienste verbessern. Die [service quality components](#) lassen sich in folgende Kategorien unterteilen:

- Verfügbarkeitskomponenten
- Zugriffskomponenten
- Verwaltungskomponenten

## Verfügbarkeitskomponenten

Verfügbarkeitskomponenten sorgen für eine nahezu kontinuierliche Systembetriebszeit von Systemdienstkomponenten und benutzerdefinierten Anwendungsdiensten. Die folgende Tabelle enthält die in Java Enterprise System enthaltenen Verfügbarkeitskomponenten und die von diesen angebotenen Dienste. Weitere Informationen zu den einzelnen Komponenten finden Sie unter „[Beschreibung der Verfügbarkeitskomponenten](#)“ auf Seite 81.



**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 enthält die in Java Enterprise System enthaltenen Zugriffskomponenten und die von diesen angebotenen Dienste. Weitere Informationen zu den einzelnen Komponenten finden Sie unter „Beschreibung der Zugriffskomponenten“ auf Seite 82.

**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 Verzeichniszugriffssteuerung und Routing für mehrere Directory Server-Instanzen.
Sun Java System Web Proxy Server	Übernimmt Cache- und Filterfunktionen sowie die Verteilung von Webinhalt sowohl für ausgehende als auch für eingehende Internetanforderungen.
Sun Java System Communications Express	Bietet, je nach Konfiguration, webbasierten Zugriff auf Messaging Server, Calendar Server und Directory Server.
Sun Java System Portal Server Secure Remote Access	Bietet sicheren Internetzugriff von außerhalb der Unternehmens-Firewall auf den Inhalt und die Dienste von Portal Server, einschließlich interner Portale.
Sun Java System Connector for Microsoft Outlook	Bietet Desktop-Clients, die Microsoft Outlook nutzen, 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 Informationen zu den einzelnen Komponenten finden Sie unter [„Beschreibung der Verwaltungskomponenten“](#) auf Seite 84.

**TABELLE 1–4** Java ES-Verwaltungskomponenten

Komponente	Bereitgestellte Verwaltungsdienste
Sun Java System Administration Server (und Console)	Bietet ein grafisches Verwaltungstool für die Konfiguration und Verwaltung von Directory Server und Messaging Server.
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.

## Gemeinsam genutzte Komponenten

Java Enterprise System enthält eine Vielzahl an lokal installierten gemeinsam genutzten Bibliotheken, von denen viele Systemdienstkomponenten und Dienstqualitätskomponenten abhängen. Die [shared components](#) von Java ES bieten lokale Dienste für Java ES-Komponenten, die auf demselben Hostcomputer ausgeführt werden.

Gemeinsam genutzte Komponenten werden oft eingesetzt, um die Portierbarkeit zwischen unterschiedlichen Betriebssystemen zu gewährleisten. Beispiele für gemeinsam genutzte Java Enterprise System-Komponenten: 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 85.

Die gemeinsam genutzten Komponenten werden automatisch vom Java ES-Installationsprogramm entsprechend der zu installierenden Systemdienste und Dienstqualitätskomponenten installiert.

---

# Arbeiten mit Java Enterprise System

Beim Erstellen von auf Java Enterprise System-Software basierenden Unternehmenslösungen fallen einige Standardaufgaben an. Diese Aufgaben fallen je nach Startpunkt der Einführung von Java Enterprise System und der Art der Lösung, die erstellt und bereitgestellt werden soll, unterschiedlich aus.

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

## Java Enterprise System-Lösungslebenszyklus

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

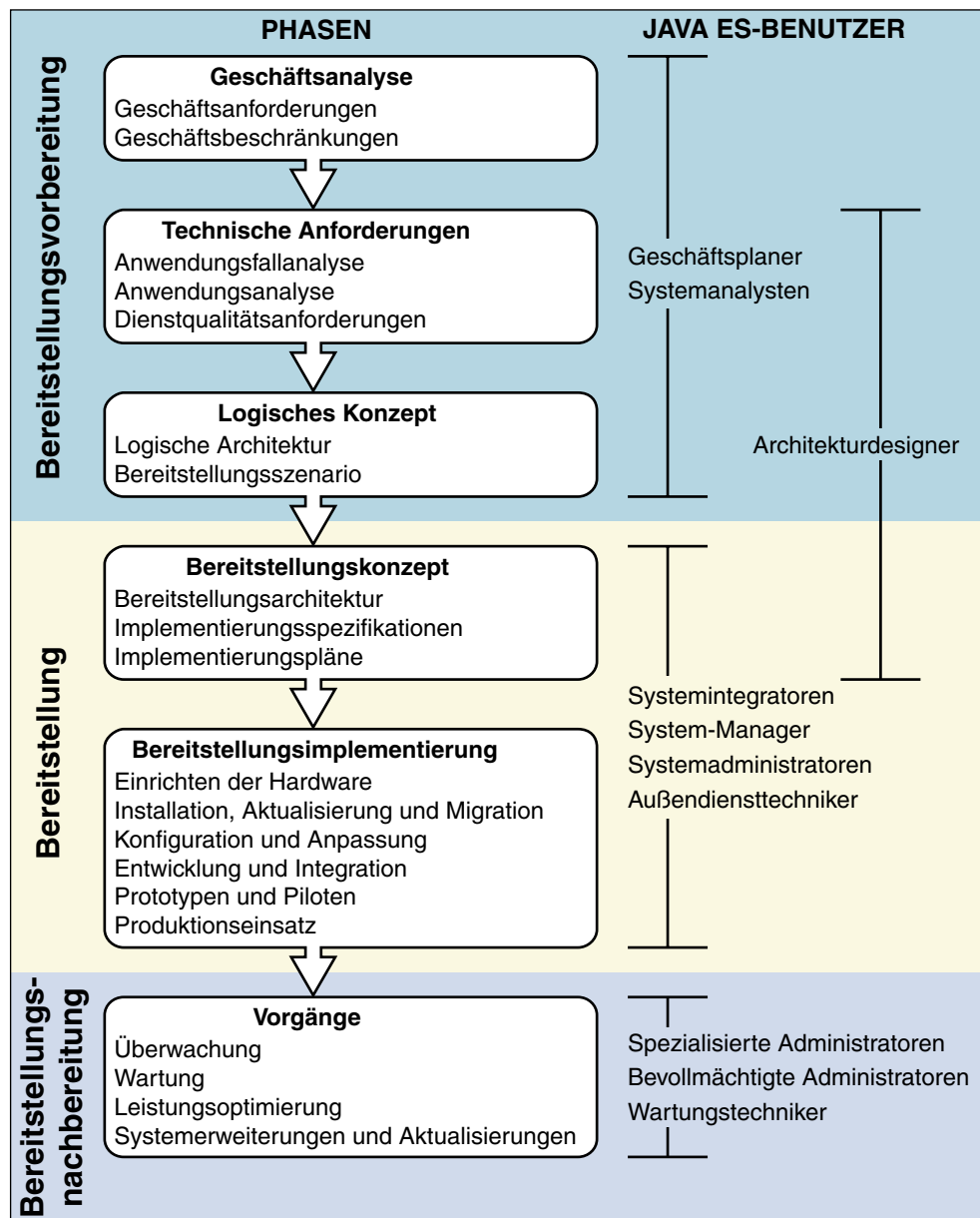


ABBILDUNG 1-3 Phasen des Lebenszyklus einer Lösung und Benutzerkategorien

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

- **Bereitstellungsvorbereitung.** In dieser Phase wird eine Unternehmensanforderung in ein Bereitstellungsszenario übertragen: 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 umgesetzt. 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 in [Kapitel 4](#) eingehend erläutert.

[Abbildung 1–3](#) zeigt die Java ES-Benutzer, die üblicherweise die Aufgaben in den einzelnen Lebenszyklusphasen durchführen. Wenn Sie mit Java ES 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 im Lebenszyklus

Benutzer	Fähigkeiten und Kenntnisse	Phasen
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

TABELLE 1-5 Java ES-Benutzerkategorien für Aufgaben im Lebenszyklus (Fortsetzung)

Benutzer	Fähigkeiten und Kenntnisse	Phasen
Systemintegrator	Sehr technisch orientiert.	Bereitstellungskonzept
Außendiensttechniker	Tiefgehende Kenntnis der IT-Umgebungen.	Bereitstellungsimplementierung
Systemadministrator	Erfahrung mit der Implementierung verteilter Softwarelösungen.	
System-Manager	Kenntnis der Netzwerkarchitektur, Protokolle, Geräte und Sicherheit. Kenntnis der Skript- und Programmiersprachen.	
Spezialisierter Systemadministrator	Spezielle technische Kenntnisse oder Produktkenntnisse.	Vorgänge
Bevollmächtigter Administrator	Gute Kenntnis der Hardware, Plattformen, Verzeichnisse und Datenbanken.	
Wartungstechniker	Geschult in der Überwachung, Fehlerbehebung und Aufrüstung von Software. Kenntnis der Systemverwaltung für Betriebssystemplattformen.	

## Java Enterprise System-Einführungsszenarien

Der Unternehmensbedarf, der zur Einführung von Java ES führt, ist sehr unterschiedlich. Das übergeordnete Ziel nahezu jeder Java ES-Bereitstellung stimmt jedoch mit einem der folgenden **adoption scenarios** überein:

- **Neues System.** Sie verfügen über kein Softwaresystem und beginnen mit der Bereitstellung der Java Enterprise System-Software, um eine neue Geschäftslösung zu unterstützen.
- **Verbesserung.** Sie verfügen über eine IT-Infrastruktur und ersetzen einige oder alle Teile Ihres Systems durch Java ES-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 verfügen über eine IT-Infrastruktur und stellen Java Enterprise System-Software bereit, die noch nicht Bestandteil Ihres Systems 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 ES-Portal oder Java-Authentifizierung und -Autorisierung für vorhandene Dienste.
- **Aufrüstung.** Sie verfügen über eine IT-Infrastruktur bestehend aus einer früheren Version von Java Enterprise System oder Sun-Produkten, die Java Enterprise System vorausgegangen sind, und nehmen eine Aufrüstung auf die aktuelle

Version der Java Enterprise System-Komponenten vor.

Jedes Einführungsszenario bietet eigene Aspekte und Herausforderungen. Der in [Abbildung 1–3](#) gezeigte Lösungslebenszyklus kommt unabhängig davon, welches Einführungsszenario auf Ihre Situation zutrifft, zur Anwendung. 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 Einführungsszenarien:

- **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 macht häufig eine Integration der neuen Softwarekomponenten in die verbleibenden Subsysteme erforderlich. Zur Integration kann die Entwicklung neuer Schnittstellenebenen, der Einsatz von J2EE Connectoren oder Ressourcen-Adaptoren, die Neukonfiguration vorhandener Komponenten sowie die Implementierung von Datentransformationsschemata gehören.
- **Schulung.** Nahezu jede Veränderung der Infrastruktur zieht Änderungen der IT-Verfahren und der erforderlichen Kenntnisse nach sich. 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 Einführungsszenario können die Hardwareressourcen zu einem wichtigen Faktor werden.

Die folgende Tabelle fasst die Art der Punkte zusammen, die bei den einzelnen Einführungsszenarien von Java ES von Bedeutung sind.

**TABELLE 1–6** Aspekte unterschiedlicher Java ES-Einführungsszenarien

Einführungsszenarien	Migration	Integration	Schulung	Hardware
Neues System	Nicht zutreffend	Relativ einfache Integration neuer Komponenten	Normalerweise ein wichtiger Aspekt	Ausgleich zwischen Ausrüstungskosten und Arbeitskosten. <sup>1</sup>

<sup>1</sup> 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.

TABELLE 1-6 Aspekte unterschiedlicher Java ES-Einführungsszenarien (Fortsetzung)

Einführungsszenarien	Migration	Integration	Schulung	Hardware
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
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 aufrüsteter Komponenten	Relativ unwesentlicher Aspekt	Relativ unwesentlicher Aspekt

## 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 von Java Enterprise System-Software, der durch die Ausgangssituation hinsichtlich der Systemsoftware und das zu erreichende Ziel gekennzeichnet ist. Es gibt vier grundlegende Java Enterprise System-Einführungsszenarien: neues System, Ersatz, Erweiterung und Aufrüstung.
- Komponente** Eine Softwarelogikeinheit, die zum Aufbau von verteilten Anwendungen verwendet wird. Bei einer Komponente kann es sich um eine der in Java Enterprise System enthaltenen [system components](#) handeln oder um eine kundenspezifische entwickelte [Anwendungskomponente](#). Eine Anwendungskomponente entspricht normalerweise einem verteilten Komponentenmodell (z. B. CORBA und der J2EE™-Plattform) und führt bestimmte Computerfunktionen durch. Diese Komponenten bieten einzeln oder kombiniert [business services](#) und können als [web services](#) 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).
<b>Endbenutzer</b>	Eine Person, die 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 <b>Bereitstellungs-architektur</b> der Anwendung.
<b>Dienst</b>	Eine Softwarefunktion, die von einem oder mehreren <b>clients</b> ausgeführt wird. Diese Funktion kann auf einer sehr niedrigen Ebene, beispielsweise eine Arbeitsspeicherverwaltung, oder auf hoher Ebene dargestellt werden, beispielsweise eine Kreditüberprüfung durch einen <b>Geschäftsdienst</b> . Ein Dienst auf hoher Ebene kann aus einer Reihe einzelner Dienste bestehen. Die Dienste können lokal (für lokale Clients verfügbar) oder verteilt (für Remote-Clients verfügbar) sein.
<b>Dienstqualitätskomponente</b>	Eine der in Java Enterprise System enthaltenen Arten von <b>system components</b> . Komponenten verbessern die Verfügbarkeit, Sicherheit, Skalierbarkeit, Zweckmäßigkeit sowie andere Bereiche der Systemdienstkomponenten und verteilten Anwendungskomponenten.
<b>Gemeinsam genutzte Komponente</b>	Eine der in Java Enterprise System enthaltenen Arten von <b>system components</b> . Gemeinsam genutzte Komponenten, normalerweise Bibliotheken, stellen für andere Systemkomponenten lokale Dienste bereit. Im Gegensatz dazu stellt eine <b>Systemdienstkomponente</b> verteilte Infrastrukturdienste für andere Systemkomponenten (oder für <b>application components</b> bereit).
<b>Systemkomponente</b>	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: <b>system service components</b> , die verteilte Infrastrukturdienste bieten, <b>service quality components</b> , die die Systemdienstkomponenten durch Zugriffs- und Verwaltungsdienste unterstützen, und <b>shared components</b> , die lokale Dienste für andere Systemkomponenten bieten.
<b>Systemdienst</b>	Einer oder mehrere verteilte <b>services</b> , die die die von einem Java Enterprise System bereitgestellten einzigartigen Funktionen definieren. Systemdienste erfordern in der Regel die Unterstützung durch eine Reihe von <b>service quality components</b> , <b>shared components</b> oder mehrere von beiden Komponentenarten.
<b>Systemdienstkomponente</b>	Eine oder mehrere der in Java Enterprise System enthaltenen Arten von <b>system components</b> . Systemdienstkomponenten bilden die wesentlichen Infrastrukturdienste von Java Enterprise System: Zugangsdienste, Kommunikations- und Zusammenarbeitsdienste, Identitäts- und Sicherheitsdienste, Web- und Anwendungsdienste

sowie Verfügbarkeitsdienste.

# Lösungsarchitekturen von Java Enterprise System

---

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

architectures von Java ES-Lösungen bieten zwei Aspekte: Eine [Logische Architektur](#) und eine [Bereitstellungs-architektur](#). 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 logischen Architekturen als auch in Bereitstellungsarchitekturen 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 35
- „Beispiel für Java Enterprise System-Lösungsarchitektur“ auf Seite 50
- „In diesem Kapitel enthaltene Schlüsselbegriffe“ auf Seite 53

---

## Java Enterprise System-Architekturframework

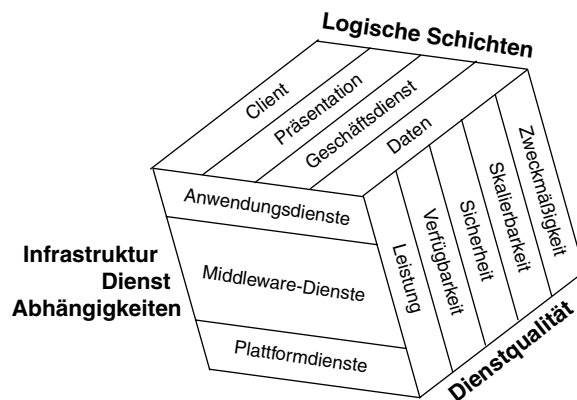
Java ES-Komponenten unterstützen die Bereitstellung verteilter Softwarelösungen für die Anforderungen von Unternehmen.

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 Lösungen 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 (Verfügbarkeit, Sicherheit, Skalierbarkeit und Zweckmäßigkeit) erreicht werden, wobei die Rolle der Dienstqualitätskomponenten berücksichtigt wird (siehe „Dienstqualitätskomponenten“ auf Seite 24).

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 (sowohl [application components](#) als auch 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

Bei der Konzeption eines verteilten Softwaresystems müssen Sie unabhängig davon, ob das Softwaresystem überwiegend aus selbst entwickelten oder fertigen Java ES-Komponenten besteht, einige Infrastrukturdienste berücksichtigen. Diese Dienste arbeiten auf vielen Ebenen.

[Abbildung 2-2](#) zeigt die Dimension der Dienstabhängigkeiten in der Lösungsarchitektur. Die in dieser Abbildung dargestellten Ebenen sind eine vergrößerte Ansicht der in [Abbildung 1-1](#) 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 derer 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.

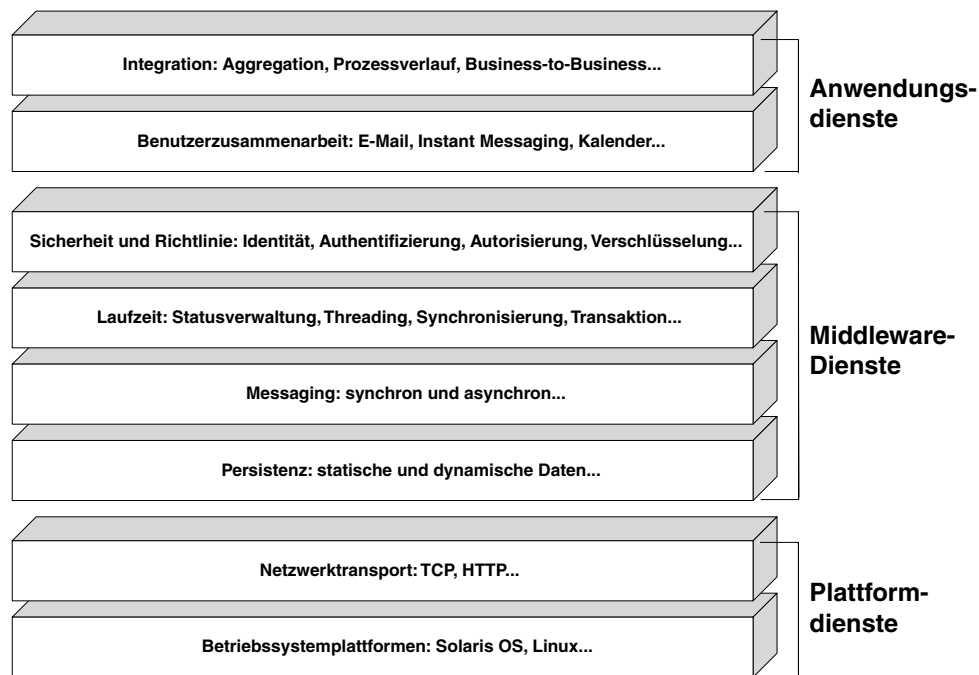


ABBILDUNG 2-2 Dimension 1: Infrastrukturdienstebenen

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 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.
- **Runtime.** 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.

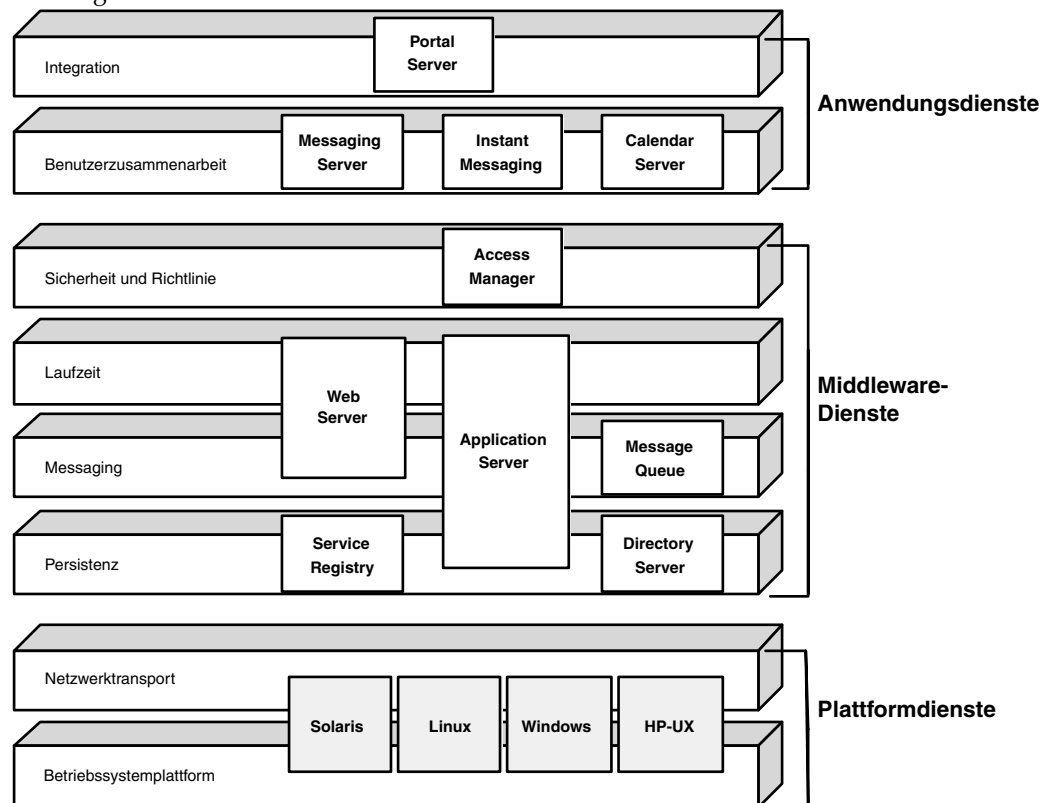


ABBILDUNG 2-3 Java ES-Systemdienstkomponenten



---

**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 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)

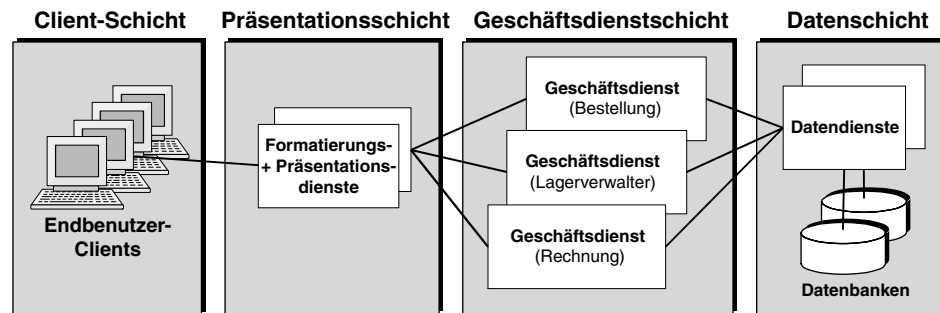
**TABELLE 2-1** Beziehungen zwischen Java ES-Systemdienstkomponenten (Fortsetzung)

Komponente	Ist abhängig von	Bietet Unterstützung für
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
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
Service Registry	Keine	Application Server-basierte Komponenten

## 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 Lösungsarchitektur 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](#) 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 [business services](#) 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. 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 [web services](#) zusammengefasst werden, die dem Schnittstellenstandard SOAP (Simple Object Access Protocol) entsprechen. Geschäftsdienste können auch als eigenständige [servers](#), 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](#) gezeigte 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.

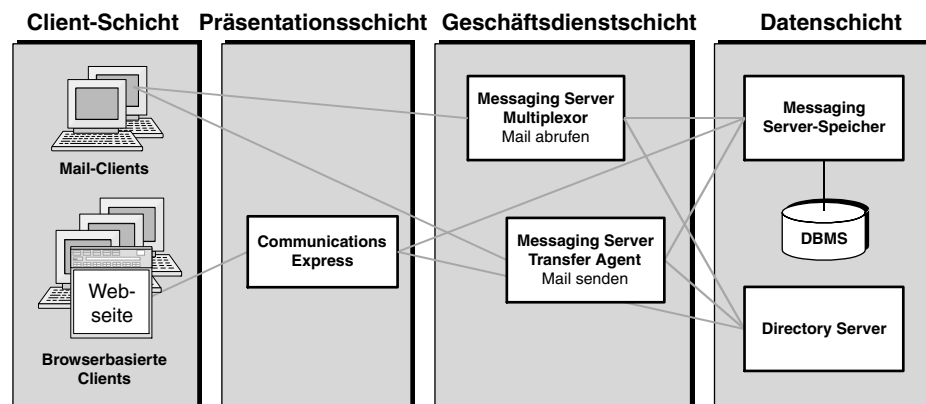
Wie Sie Anwendungs- oder Infrastrukturkomponenten auf eine Hardwareumgebung (Ihre Bereitstellungsarchitektur) abbilden, hängt von vielen Faktoren ab, die durch die Größe und Komplexität Ihrer Softwarelösung bestimmt

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](#) gezeigt, 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.

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 folgenden Abbildung zeigt, als separate Komponenten dargestellt, die sich auf unterschiedlichen logischen Schichten befinden.



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

---

**Hinweis –** [Abbildung 2-5](#) soll keine vollständige logische Architektur darstellen. 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. Durch die physische Trennung wird eine größere Flexibilität bei der Einhaltung der Dienstqualitätsanforderungen ermöglicht (siehe „[Dimension 3: Dienstqualität](#)“ auf Seite 46). 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).

**TABELLE 2-2** Dienstqualitäten mit Auswirkung auf die Lösungsarchitektur (Fortsetzung)

Systemdienstqualitäten	Beschreibung
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, der Behebung 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 24 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
High Availability Session Store	Verfügbarkeit
Portal Server Secure Remote Access	Sicherheit Skalierbarkeit
Sun Cluster	Verfügbarkeit Skalierbarkeit
Web Proxy Server	Sicherheit Leistung Zweckmäßigkeit

## Sun Cluster-Software

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

Ein Cluster besteht aus lose miteinander verbundenen Computern, die zusammen eine einzelne Client-Ansicht der Dienste, Systemressourcen und Daten bieten. 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.

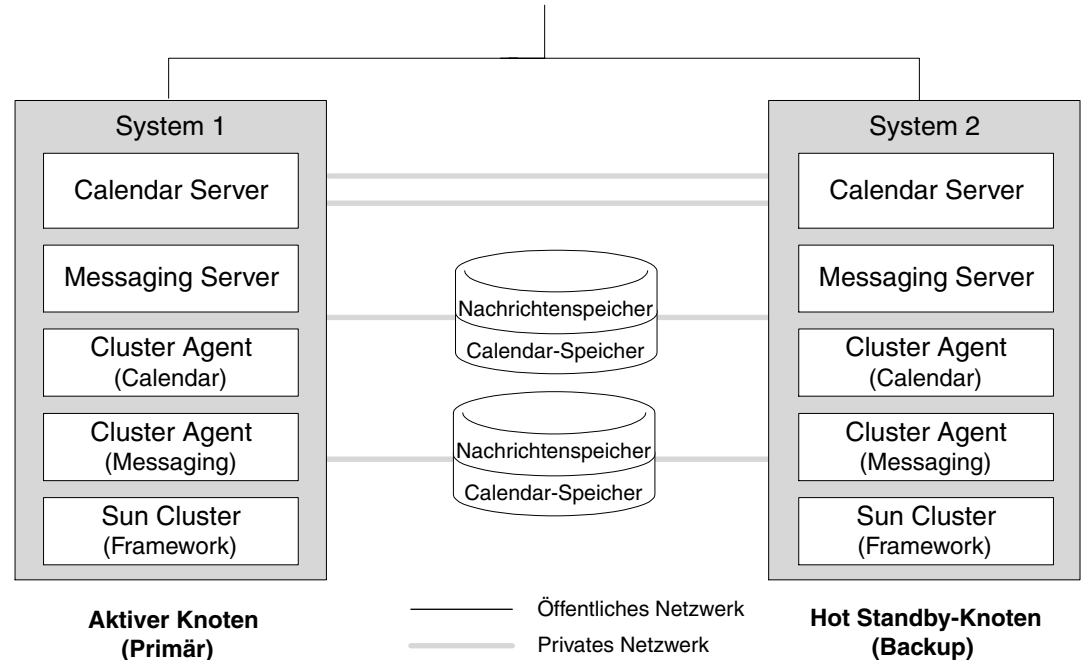


ABBILDUNG 2-6 Verfügbarkeitskonzept mit Sun Cluster-Knoten

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. Daher muss die Sun Cluster-Software auf jedem Computer in einem Cluster installiert werden.

## Synthese der drei Architekturdimensionen

Zusammengenommen 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.

Aufgrund 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.

Eine detaillierte Beschreibung der Entwurfsmethodik für die Verwendung des unter [„Java Enterprise System-Architekturframework“](#) auf Seite 35 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

Unternehmensanforderung	Beschreibung	Erforderliche Java ES-Dienste
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

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

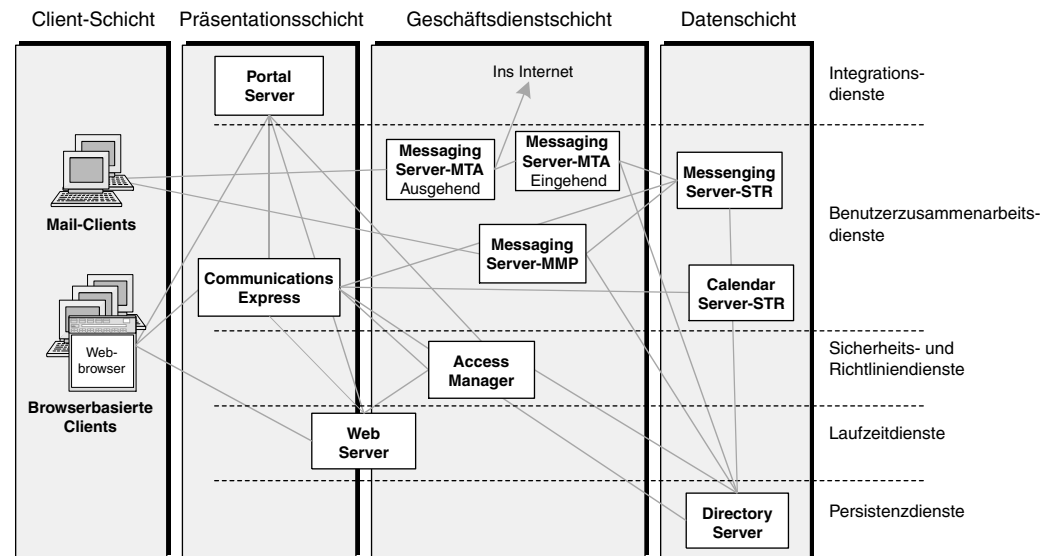


ABBILDUNG 2-7 Logische Architektur des Unternehmenskommunikationsszenarios

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 hinsichtlich der Authentifizierung und Autorisierung über Single Sign-On für den Portal Server und andere webbasierte Komponenten der Präsentationsschicht. 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 *Sun Java Enterprise System 2005Q4 Handbuch zur Bereitstellungsplanung*.

## 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 *Sun Java Enterprise System 2005Q4 Handbuch zur Bereitstellungsplanung*.

---

## 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.

<b>Anwendungs-komponente</b>	Eine kundenspezifisch entwickelte Software-Komponente, die einige spezifische Datenverarbeitungsfunktionen durchführt und <b>end users</b> oder andere Anwendungskomponenten mit <b>business services</b> versorgt. Eine Anwendungskomponente entspricht normalerweise einem verteilten Komponentenmodell (z. B. CORBA und der J2EE™-Plattform). Diese Komponenten können (einzeln oder kombiniert) als <b>web services</b> zusammengefasst werden.
<b>Architektur</b>	Ein Konzept, das die logischen und physischen modularen Blöcke einer verteilten Anwendung (oder eines anderen Softwaresystems) sowie ihre Beziehungen untereinander darstellt. Für eine <b>Verteilte</b>

	<p><b>Unternehmens- anwendung</b> umfasst das Architekturkonzept im Allgemeinen sowohl die <b>Logische Architektur</b> als auch die <b>Bereitstellungs- architektur</b> der Anwendung.</p>
<b>Geschäftsdienst</b>	<p>Eine <b>Anwendungs- komponente</b> oder 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 <b>Webdienst</b> zusammengefasst sind, oder ein eigenständiger <b>Server</b> sein.</p>
<b>Client</b>	<p>Eine Software, die Software- <b>services</b> anfordert. (Hinweis: Es handelt sich hierbei nicht um eine Person – siehe <b>Endbenutzer</b>.) Ein Client kann ein Dienst sein, der einen anderen Dienst anfordert, oder eine GUI-Komponente, auf die ein Endbenutzer zugreift.</p>
<b>Bereitstellungs- architektur</b>	<p>Ein starkes Konzept, das die <b>Logische Architektur</b> einer physischen Computerumgebung zuordnet. Die physische Umgebung umfasst die Computer in einer Intranet- oder Internetumgebung, die Netzwerkverbindungen zwischen ihnen sowie andere physische Geräte, die zur Unterstützung der Software erforderlich sind.</p>
<b>Logische Architektur</b>	<p>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 <b>application components</b> als auch die Infrastrukturdienste, die für deren Unterstützung erforderlich sind.</p>
<b>Server</b>	<p>Ein Softwarevorgang mit mehreren Threads (im Gegensatz zu einem Hardwareserver), der einen verteilten <b>service</b> oder eine geschlossene Gruppe von Diensten für <b>clients</b> bereitstellt, die über eine externe Schnittstelle auf den Dienst zugreifen</p>
<b>Webdienst</b>	<p>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).</p>

# 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 55
- „Integrierte Identitäts- und Sicherheitsdienste“ auf Seite 58
- „In diesem Kapitel enthaltene Schlüsselbegriffe“ auf Seite 62

---

## 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.

Bei dem 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 Java ES-Installationsprogramms behandelt (ausführliche Informationen finden Sie im *Sun Java Enterprise System 2005Q4 Installationshandbuch für UNIX*):

- „Überprüfung bereits vorhandener Software“ auf Seite 56
- „Überprüfung der Abhängigkeiten“ auf Seite 56
- „Erstkonfiguration“ auf Seite 57
- „Deinstallation“ auf Seite 57

## 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.

Das Installationsprogramm überprüft außerdem, 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 festzustellen, ob 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

Java Enterprise System bietet 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 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“ auf Seite 58
- „Authentifizierung und Single Sign-On“ auf Seite 59

## Einzelidentität

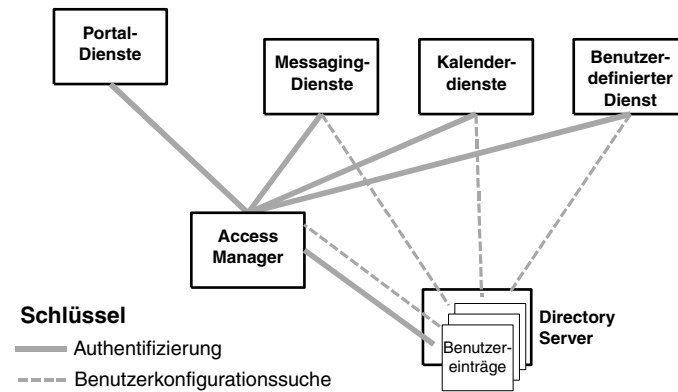
Innerhalb einer Java ES-Umgebung verfügt jeder Benutzer über eine einzelne integrierte Identität. Anhand dieser [Einzelidentität](#) kann dem Benutzer der Zugriff auf verschiedene Ressourcen gestattet werden, beispielsweise Portale, Webseiten und Dienste wie Nachrichtendienste, Kalenderdienste und Instant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

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

Der Benutzerzugriff auf einen Dienst oder 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 Java Enterprise System-Funktion 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 außerdem 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 an die gesicherten Ressourcen im Namen von Access Manager 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 7 2005Q4“ auf Seite 76).

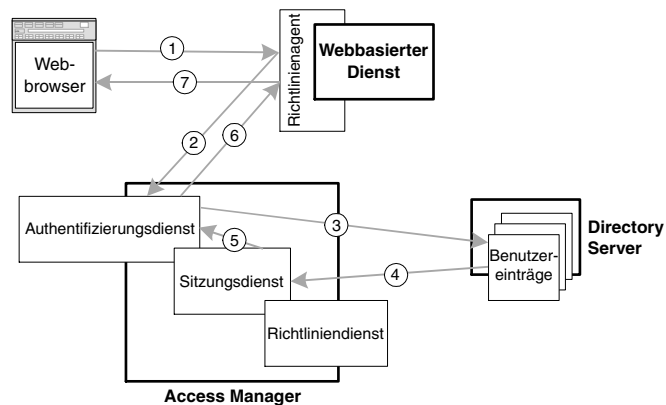
## Authentifizierung

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 Authentifizierungsanfrage 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 mit dem 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.

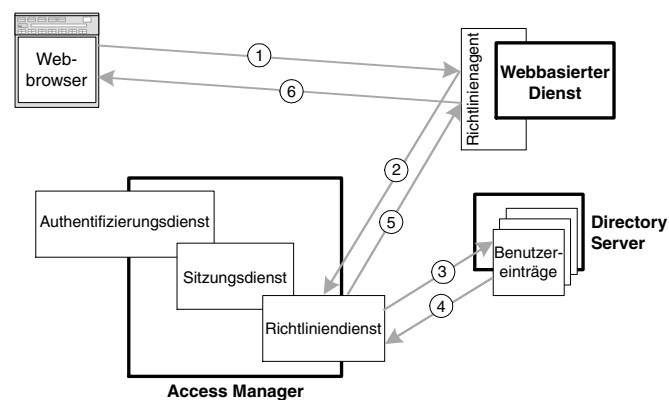


ABBILDUNG 3-3 Autorisierungssequenz

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 die 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 zu verdeutlichen, welche Beziehung zwischen diesen Begriffen im Java Enterprise System-Kontext vorliegt.

<b>Verzeichnis</b>	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.
<b>Richtlinie</b>	Eine Richtlinie ist 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.
<b>Einzelidentität</b>	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.
<b>Single Sign-On</b>	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

---

In diesem Kapitel werden die Konzepte und die dazugehörige Terminologie besprochen, die für die einzelnen Phasen des Java ES-Lösungslebenszyklus relevant sind. 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 65
- „Bereitstellung“ auf Seite 66
- „Bereitstellungsnachbereitung“ auf Seite 72
- „In diesem Kapitel enthaltene Schlüsselbegriffe“ auf Seite 72

---

## Aufgaben des Lösungslebenszyklus

Der Lösungslebenszyklus wurde in [Kapitel 1](#) als ein Standardansatz für die Implementierung von Geschäftslösungen mit der Java ES-Software vorgestellt. In diesem Kapitel werden die Aufgaben beschrieben, die in den einzelnen Phasen des Lebenszyklus anfallen. Das Lebenszyklusdiagramm wird zu Referenzwecken in [Abbildung 4-1](#) noch einmal wiederholt.

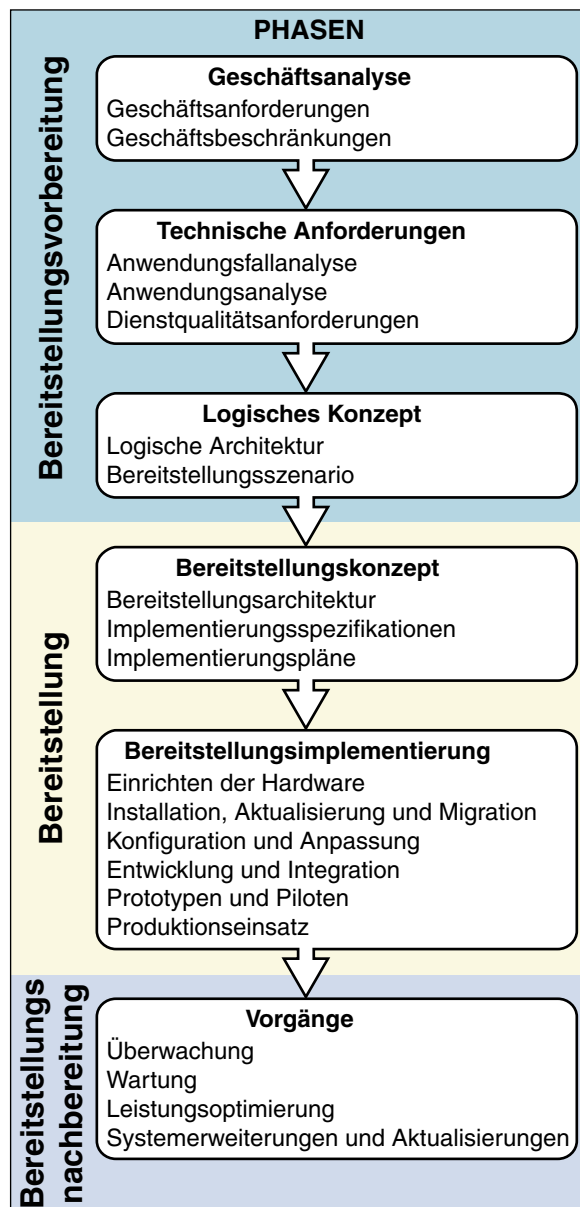


ABBILDUNG 4-1 Aufgaben des Lösungslebenszyklus



---

## Bereitstellungsvorbereitung

In den Phasen der **Bereitstellungs- vorbereitung** des Lebenszyklus übertragen Sie die Analyse der Geschäftsanforderungen in ein **Bereitstellungsszenario**. Das Bereitstellungsszenario dient als Spezifikation für ein Bereitstellungskonzept.

Die Aufgaben der Bereitstellungsvorbereitung werden in drei Phasen aufgeteilt, wie in **Abbildung 4–1** gezeigt:

- **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 verwenden Sie die Ergebnisse der Geschäftsanalyse, um **use cases** zu erstellen, 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**), 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 in **Kapitel 2** 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 zusammen mit Dienstqualitätsanforderungen wie Leistung, Verfügbarkeit und Sicherheit in einem Bereitstellungsszenario zusammengefasst, wie in der folgenden Abbildung gezeigt. Weitere Informationen zu den Phasen der Bereitstellungsvorbereitung des Lebenszyklus finden Sie im *Sun Java Enterprise System 2005Q4 Handbuch zur Bereitstellungsplanung*.

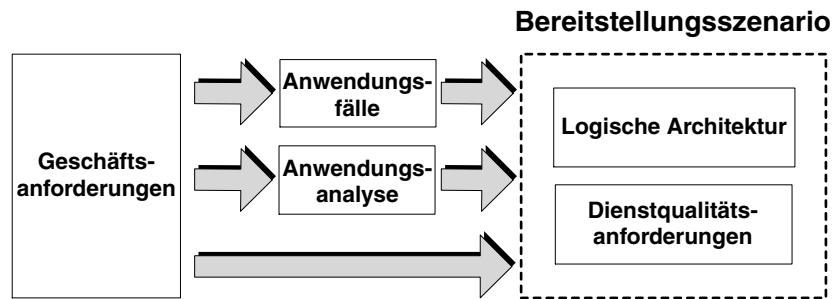


ABBILDUNG 4-2 Spezifizieren eines Bereitstellungsszenarios

## Bereitstellung

In den Phasen der **Bereitstellung** des Lebenszyklus übertragen Sie ein Bereitstellungsszenario in ein Bereitstellungskonzept, das anschließend implementiert, getestet und in einer Produktionsumgebung eingesetzt wird.

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

Die Aufgaben der Bereitstellung werden in zwei Phasen aufgeteilt, wie in **Abbildung 4-1** gezeigt:

- „**Bereitstellungskonzept**“ auf Seite 67. 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**“ auf Seite 69. 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.

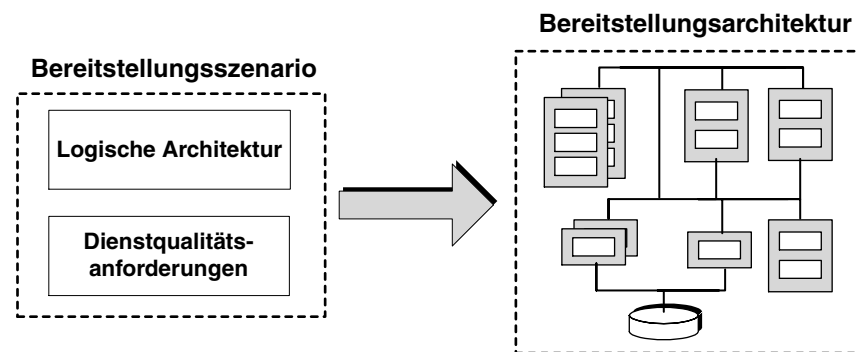
## Bereitstellungskonzept

In der Bereitstellungskonzeptphase 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- Bereitstellungs- architektur n**.

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 Bereitstellungszenarios, 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:

- Hardware-Setup

- 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 Angaben zur ausführlichen Dokumentation dieser Aufgaben finden Sie im *Übersicht über die Dokumentation zu Sun Java Enterprise System 2005Q4*.

## 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 55).

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 58).

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 Portal Server an, damit dieser Zugangskanäle bereitstellt, 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 für die 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 Geschäftsdienstsicht 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. 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 einsetzen (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 **Bereitstellungs- nachbereitung** führen Sie eine bereitgestellte Lösung in einer Produktionsumgebung aus. 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 das Ersetzen von Nicht-Java ES-Komponenten. 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 Administrationstools für ihre 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.

<b>Bereitstellung</b>	Ein Abschnitt im Lebenszyklus einer Java Enterprise System-Lösung, in dem ein Bereitstellungsszenario in ein Bereitstellungskonzept übertragen, dann implementiert, als Prototyp getestet und schließlich
-----------------------	---



in einer Produktionsumgebung eingesetzt wird. Das Endprodukt dieses Prozesses wird ebenfalls als Bereitstellung (oder bereitgestellte Lösung) bezeichnet.

<b>Bereitstellungsszenario</b>	Eine <b>Logische Architektur</b> für eine Java Enterprise System-Lösung und die Dienstqualitätsanforderungen, die von der Lösung erfüllt werden müssen, um den Geschäftsanforderungen zu entsprechen. Die Dienstqualitätsanforderungen betreffen u.a. folgende Aspekte: Leistung, Verfügbarkeit, Sicherheit, Zweckmäßigkeit und Skalierbarkeit/latente Kapazität. Ein Bereitstellungsszenario ist der Ausgangspunkt für ein Bereitstellungs-konzept.
<b>Entwicklung</b>	Eine Aufgabe im Bereitstellungsprozess der Java Enterprise System-Lösung, bei der die angepassten Komponenten einer <b>Bereitstellungs-architektur</b> programmiert und getestet werden.
<b>Bereitstellungsvorbereitung</b>	Ein Abschnitt im Lebenszyklus der Java Enterprise System-Lösung, in dem die Geschäftsanforderungen in ein <b>Bereitstellungsszenario</b> umgesetzt werden: Eine <b>Logische Architektur</b> und eine Reihe von Dienstqualitätsanforderungen, die die Lösung erfüllen muss.
<b>Bereitstellungsnachbereitung</b>	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.
<b>Referenz-Bereitstellungsarchitektur</b>	Eine <b>Bereitstellungs-architektur</b> , die hinsichtlich der Leistung entworfen, implementiert und getestet wurde. Referenz-Bereitstellungsarchitekturen dienen als Ausgangspunkt für die Konzipierung von Bereitstellungsarchitekturen für individuell angepasste Lösungen.
<b>Anwendungsfall</b>	Eine bestimmte Endbenutzeraufgabe oder eine Reihe von Aufgaben, die eine <b>Verteilte Unternehmens-anwendung</b> ausführt und die als Basis für die Konzipierung, das Testen und das Messen der Leistung der Anwendung dient.



## Referenzliste: Java Enterprise System-Komponenten

---

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

- „Beschreibungen der Systemdienstkomponenten“ auf Seite 76. Diese Komponenten bieten wesentliche Java ES-Infrastrukturdienste, die für die Unterstützung verteilter Unternehmensanwendungen benötigt werden. Zu diesen Diensten gehören, wie unter „Wozu benötigen Sie Java Enterprise System?“ auf Seite 19 beschrieben, Zugangsdienste, Kommunikations- und Zusammenarbeitsdienste, Identitäts- und Sicherheitsdienste, Web- und Anwendungsdienste sowie Verfügbarkeitsdienste.
- „Beschreibung der Dienstqualitätskomponenten“ auf Seite 81. 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“ auf Seite 85. Diese Komponenten sind lokale Bibliotheken, die von allen Java ES-Komponenten, die auf bestimmten Hostcomputern ausgeführt werden, gemeinsam genutzt werden können.

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

Eine Übersicht über die Dokumentation zu den verschiedenen Komponenten finden Sie im *Übersicht über die Dokumentation zu Sun Java Enterprise System 2005Q4*.

---

## 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 7 2005Q4“ auf Seite 76
- „Sun Java System Application Server Enterprise Edition 8.1 2005Q4“ auf Seite 77
- „Sun Java System Calendar Server 6 2005Q4“ auf Seite 77
- „Sun Java System Directory Server 5 2005Q4“ auf Seite 78
- „Sun Java System Instant Messaging 7 2005Q4“ auf Seite 78
- „Sun Java System Message Queue 3.6 2005Q4“ auf Seite 78
- „Sun Java System Messaging Server 6 2005Q4“ auf Seite 79
- „Sun Java System Portal Server 6 2005Q4“ auf Seite 80
- „Sun Java System Service Registry 3 2005Q4“ auf Seite 80
- „Sun Java System Web Server 6.1 2005Q4“ auf Seite 80

### Sun Java System Access Manager 7 2005Q4

Sun Java System Access Manager (Access Manager) bietet Organisationen eine Infrastruktur für die Verwaltung von Vorgängen, die bei der Verwaltung digitaler Identitäten von Kunden, Angestellten und Partnern anfallen, die die webbasierten Dienste und nicht webbasierten Anwendungen der Organisation nutzen. 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.** Mit dieser Unterkomponente werden Benutzeridentitäten erstellt und verwaltet sowie Richtlinien definiert und ausgewertet, die basierend auf der Identität der Benutzer Zugriff auf die Java ES-Ressourcen bieten. Diese Unterkomponente enthält außerdem die Unterkomponenten Access Manager SDK und Delegated Administrator (siehe „Sun Java System Delegated Administrator 6 2005Q4“ auf Seite 85).
- **Access Manager SDK.** Bietet eine Remote-Schnittstelle für Access Manager. Diese Unterkomponente muss auf jedem Computer installiert werden, der als Host für eine Java ES-Komponente dient, auf die Access Manager remote zugreift.
- **Access Manager Administration Console.** Hierbei handelt es sich um eine grafische Benutzeroberfläche, die Identitätsdienste und Richtlinienverwaltung zusammenfasst und eine einzige Oberfläche liefert, die Benutzern das Erstellen

und Verwalten von Benutzerkonten, Service-Attributen und Zugangsregeln im Directory Server ermöglicht.

- **Common Domain Services for Federation Management.** Ermöglicht Benutzern die Verwendung einer einzelnen Identität, um auf Anwendungen zuzugreifen, die von mehreren angegliederten Diensteanbietern angeboten werden.

## Sun Java System Application Server Enterprise Edition 8.1 2005Q4

Sun Java System Application Server (Application Server) bietet eine J2EE-kompatible Plattform für die Entwicklung und Bereitstellung von Web- und Anwendungsdiensten. Application Server bietet die Infrastrukturdienste für die Interaktion zwischen eng verknüpften und verteilten Komponenten, einschließlich Remote-Methodenaufruf und anderer 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 Verwaltungsclients für die Verwaltung und Konfiguration von Application Server-Installationen und gehostete Anwendungen. Der Administration Client hilft auch bei der Bereitstellung von Anwendungen.
- **Command Line Administration Tool.** Bietet befehlszeilengesteuerte Verwaltungsclients für die Verwaltung und Konfiguration von Application Server-Installationen und gehostete Anwendungen. 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 2005Q4

Sun Java System Calendar Server (Calendar Server) ist eine skalierbare, webbasierte Lösung für die zentrale Kalenderverwaltung und Terminplanung für Unternehmen und Dienstleister. Calendar Server unterstützt persönliche Kalender und Gruppenkalender sowie Kalender für Ressourcen wie Konferenzräume und Ausrüstung.

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

## Sun Java System Directory Server 5 2005Q4

Sun Java System Directory Server (Directory Server) bietet einen zentralen Verzeichnisdienst für Ihre Intranet-, Netzwerk- und Extranet-Informationen. Directory Server wird in bestehende Systeme integriert und fungiert als zentraler Ablageort 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 2005Q4

Sun Java System Instant Messaging (Instant Messaging) ermöglicht Endbenutzern die Teilnahme an Instant Messaging- und Chat-Sitzungen, das Senden von Warnmeldungen untereinander und die Verbreitung aktueller Gruppen-News in Echtzeit. Instant Messaging eignet sich sowohl für das Intranet als auch für das Internet und unterstützt die Kommunikation mit anderen Messaging-Anbietern.

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 Multiplexor-Software.
- **Instant Messaging Resources.**
- **Access Manager Instant Messaging Service.**

## Sun Java System Message Queue 3.6 2005Q4

Sun Java System Message Queue (Message Queue) ist eine auf Standards basierende Lösung für die häufig problematische Kommunikation zwischen Anwendungen und verlässlicher Nachrichtenweiterleitung. Message Queue ist ein Messaging-System für Unternehmen, das den Java Message Service (JMS) Open 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 Ausgaben erhältlich:

- **Enterprise Edition** (Standard). }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 installierbare Einzelkomponenten zur Verfügung.

## Sun Java System Messaging Server 6 2005Q4

Sun Java System Messaging Server (Messaging Server) ist ein leistungsfähiger, auf Standards basierender Internet Messaging Server sowohl für Unternehmen als auch 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:

- **Nachrichtenspeicher.** Speichert Nachrichten und ruft Nachrichten ab.
- **Message Transfer Agent (MTA).** Unterstützt das Senden von E-Mails durch das Bearbeiten von SMTP-Verbindungen, die Weiterleitung von E-Mails und die Zustellung von Nachrichten an die richtigen Nachrichtenspeicher. Kann so konfiguriert werden, dass E-Mails an interne Speicher (Inbound) oder an externe E-Mail-Speicher (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 2005Q4

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

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

## Sun Java System Service Registry 3 2005Q4

Sun Java System Service Registry (Service Registry) ist ein Repository, das sowohl als Webdienst-(UDDI-)Registrierung als auch als Enterprise Business XML-(ebXML-)Registrierung für die Unterstützung von Service-Oriented Architecture-(SOA-)Anwendungen dient. Die UDDI-Registrierung wird für die Registrierung und das Auffinden von Webdiensten verwendet und die ebXML-Registrierung für das Speichern und Verwalten der Informationsartefakte, die für die Unterstützung der Unternehmensprozessintegration erforderlich sind. Zu diesen Artefakten gehören Metadaten, wie XML-Schemata, Unternehmensprozessregeln, Webdienstzugriffssteuerung, Versionskontrolle, Klassifizierungsschemata usw.

## Sun Java System Web Server 6.1 2005Q4

Sun Java System Web Server (Web Server ) ist ein sicherer auf offenen Standards basierender Multiprozess- und Multithread-Webserver. Web Server bietet Unternehmen jeder Größe hohe Leistung, Zuverlässigkeit, Skalierbarkeit sowie Verwaltungsfunktionen. Web Server unterstützt zahlreiche Softwarestandards, darunter 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 8/05 und Sun Cluster Agents“](#) auf Seite 81
- [„High Availability Session Store 2005Q4“](#) auf Seite 82

## Sun Cluster 3.1 8/05 und Sun Cluster Agents

Sun Cluster-Software bietet Hochverfügbarkeitsdienste und Skalierbarkeit für Java Enterprise System sowie für auf der Java ES-Infrastruktur basierende Anwendungen.

Ein Cluster besteht aus lose miteinander verbundenen Computern (Cluster-Knoten), die zusammen eine einzelne Client-Ansicht der Dienste, Systemressourcen und Daten bieten. Intern verwendet der Cluster redundante Computer, Interconnects, Datenspeicher und Netzwerkschnittstellen zur Bereitstellung der Hochverfügbarkeit für clusterbasierte Dienste und Daten. Sun Cluster-Software überwacht fortlaufend den Zustand der Mitglieds-knoten und anderer Clusterressourcen und verwendet die interne Redundanz zur Bereitstellung eines beinahe unterbrechungsfreien Zugriffs auf diese Ressourcen, selbst wenn Fehler auftreten.

Das Java ES-Installationsprogramm stellt Sun Cluster Core und Sun Cluster Agents als installierbare Einzelkomponenten zur Verfügung. Weitere Sun Cluster Agents sind auf separaten CDs erhältlich.

## High Availability Session Store 2005Q4

Sun Java System High Availability Session Store (HADB) bietet einen Datenspeicher, mit dem Anwendungsdaten selbst bei einem Ausfall verfügbar gemacht werden können. 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. In diesem Abschnitt werden die folgenden Java ES-Zugriffskomponenten beschrieben:

- „Sun Java System Communications Express 6 2005Q4“ auf Seite 82
- „Sun Java System Web Proxy Server 4.0.1 2005Q4“ auf Seite 83
- „Sun Java System Connector for Microsoft Outlook 6 2005Q4“ auf Seite 83
- „Sun Java System Directory Proxy Server 5 2005Q4“ auf Seite 83
- „Sun Java System Portal Server Secure Remote Access 6 2005Q4“ auf Seite 84

## Sun Java System Communications Express 6 2005Q4

Sun Java System Communications Express (Communications Express) bietet einen integrierten webbasierten Client für Kommunikation und Zusammenarbeit, der Endbenutzern eine Webschnittstelle für Kalender-, E-Mail- und Adressbuchdienste zur Verfügung stellt. Communications Express besteht aus drei Client-Modulen: Calendar, Address Book und Mail. Für den Zugriff auf Messaging Server oder Calendar Server bzw. für den Zugriff auf beide Komponenten konfigurierbar; Communications Express arbeitet entweder mit Sun Java System LDAP-Schema Version 1 (Schema 1) oder Schema 2.

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

## Sun Java System Web Proxy Server 4.0.1 2005Q4

Sun Java System Web Proxy Server (Web Proxy Server) bietet Cache-, Filter- und Verteilungsfunktionen für Webinhalte. Web Proxy Server wird häufig innerhalb von Firewalls eingesetzt, um die Anzahl der Anforderungen für remote Content Server zu reduzieren; außerhalb von Firewalls bietet Web Proxy Server einen sicheren Gateway für eingehende Internetanforderungen.

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

## Sun Java System Connector for Microsoft Outlook 6 2005Q4

Sun Java System Connector for Microsoft Outlook ermöglicht die Nutzung von Outlook Desktop-Client mit 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 es 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-Nachrichten von Messaging Server und Kalenderinformationen von Calendar Server.

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

## Sun Java System Directory Proxy Server 5 2005Q4

Sun Java System Directory Proxy Server (Directory Proxy Server) ist eine wesentliche Komponente für unternehmenskritische Verzeichnisdienste in e-Commerce-Lösungen. Directory Proxy Server ist ein LDAP-(Lightweight Directory Access Protocol-)Protokollgateway der Anwendungsschicht, das verbesserte Verzeichniszugriffssteuerung, Schemakompatibilität und Hochverfügbarkeit durch Lastenausgleich und Failover in der Anwendungsschicht zur Verfügung stellt.

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

## Sun Java System Portal Server Secure Remote Access 6 2005Q4

Sun Java System Portal Server Secure Remote Access (Portal Server Secure Remote Access) ist eine Erweiterung für Portal Server und bietet browserbasierten sicheren Remote-Zugriff auf Portal Server-Content und -Dienste von einem beliebigen Remote-Browser aus. Dadurch wird der Einsatz von Client-Software überflüssig. Durch die Integration in Portal Server wird gewährleistet, dass die Benutzer auf sichere Weise auf den Content und die Dienste zugreifen können, für die sie zugriffsberechtigt sind.

Das Java ES-Installationsprogramm stellt Portal Server Secure Remote Access als installierbare Einzelkomponenten 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 Internet-Webseiten.

## Beschreibung der Verwaltungskomponenten

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

- „Sun Java System Administration Server (und Console) 5 2005Q4“ auf Seite 85
- „Sun Java System Directory Preparation Tool 2005Q4“ auf Seite 85
- „Sun Java System Delegated Administrator 6 2005Q4“ auf Seite 85

## Sun Java System Administration Server (und Console) 5 2005Q4

Sun Java System Administration Server (Administration Server) und Server Console bieten zusammen ein grafisches Tool für die Verwaltung von Directory Server, Directory Proxy Server und Messaging Server. 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 zusammen als installierbare Einzelkomponente zur Verfügung.

## Sun Java System Directory Preparation Tool 2005Q4

Das Sun Java System Directory Preparation Script wird für die Konfiguration von Directory Server mit Schema 2 verwendet, um Messaging Server und Calendar Server mit Benutzern zu versorgen.

## Sun Java System Delegated Administrator 6 2005Q4

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

Delegated Administrator wird automatisch zusammen mit Access Manager installiert.

---

## 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 die gemeinsam genutzten Komponenten, die für die Unterstützung anderer auf einem Host-Computer installierten Java ES-Komponenten erforderlich sind.

Java Enterprise System enthält folgende gemeinsam genutzte Komponenten:

- ANT (Jakarta ANT Java/XML-basiertes Build-Tool)
- Apache Commons Logging
- Berkeley DB
- Common Agent Container
- ICU (International Components for Unicode)
- J2SE™ (Java 2 Platform, Standard Edition)
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java-API for XML-based 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
- MA (Mobile Access) Core
- MFWK (Java ES Monitoring Framework)
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- Sun Explorer Data Collector (nur Solaris)
- Sun Java Web Console
- WSCL (Web Services Common Library)

# Index

---

## A

Abhängigkeiten, 41-42, 57  
Abhängigkeitsprüfung,  
  Installationsprogramm, 56-57  
Access Manager  
  als Infrastrukturdienst, 40-41  
  als Systemdienstkomponente, 23  
  Beschreibung, 76  
Administration Server  
  als Dienstqualitätskomponente, 26  
  Beschreibung, 85  
Ant (Jakarta ANT Java/XML-basiertes  
  Build-Tool), 86  
Anwendungsdienste, 21, 37  
Anwendungen  
  Unternehmen  
    *Siehe* Verteilte  
    Unternehmensanwendungen  
  verteilt  
    *Siehe* Verteilte  
    Unternehmensanwendungen  
Anwendungsfälle  
  Definition, 73  
  Einführung, 65  
Anwendungskomponenten  
  Definition, 53  
  in logischen Schichtenarchitekturen, 43  
Apache SOAP Runtime, 86  
Application Server  
  als Infrastrukturdienst, 40-41  
  als Systemdienstkomponente, 23  
  Beschreibung, 77

## Architektur

Bereitstellung, 67-68  
Definition, 53  
Dimensionen  
  *Siehe* Architekturdimensionen  
Einführung, 35  
Lösung, 36  
Referenz, 68  
Architekturdimensionen  
  Dienstqualität, 46-49  
  Infrastrukturdienstabhängigkeiten, 37  
  logische Schichten, 42  
  Synthese, 50  
Aufgaben, Java Enterprise System, 27, 63  
Authentifizierung, 59-60  
Autorisierung, 61-62

## B

Benutzer einrichten, 68  
Benutzereinrichtung, 68  
Benutzereintrag, 58  
Benutzerkategorien  
  Architekt, 29  
  Bevollmächtigter Administrator, 30  
  Einsatztechniker, 30  
  Geschäftsplaner, 29  
  IT-Manager, 30  
  Spezialisierte Systemadministrator, 30  
  Systemadministrator, 30  
  Systemanalyst, 29  
  Systemintegrator, 30

- Benutzerprofile, 29
- Benutzerzusammenarbeitsdienste, 39
- Bereitstellung, Architektur, 67
- Bereitstellung
  - Architektur, 67-68
  - Definition, 72
  - Entwicklung und Anpassung, 70
  - Implementierung, 69-71
  - Konzept, 67-69
  - Lebenszyklusabschnitt, 66-71
  - Produktionseinsatz, 71
  - Prototyp-Tests, 70
  - Referenzarchitekturen, 68
  - Szenarien
    - Siehe* Bereitstellungsszenarien
- Bereitstellungsarchitekturen
  - Beziehung zu Schichtenarchitekturen, 44
  - Definition, 54
  - Einführung, 35
  - Entwurf von, 67-68
- Bereitstellungsnachbereitung
  - Definition, 73
  - Phasen des Lebenszyklus, 72
- Bereitstellungsszenarien
  - Definition, 73
  - Einführung, 65
- Bereitstellungsvorbereitung
  - Definition, 73
  - Phasen des Lebenszyklus, 65
- Berkeley DB, 86
- Betriebssystemdienste, 38

## C

- Calendar Server
  - als Infrastrukturdienst, 40-41
  - als Systemdienstkomponente, 23
  - Beschreibung, 77
- CAPI (Java Calendar API), 86
- Clients
  - Definition, 54
  - Systemdienstkomponenten und, 22
- Cluster
  - Siehe* Sun Cluster
- Common Agent Container, 86
- Communications Express
  - als Dienstqualitätskomponente, 25

- Communications Express (Fortsetzung)
  - Beschreibung, 82
- Connector for Microsoft Outlook
  - als Dienstqualitätskomponente, 25
  - Beschreibung, 83

## D

- Deinstallationsprogramm, 57
- Delegated Administrator
  - als Dienstqualitätskomponente, 26
  - Beschreibung, 85
- deployment, phases of life cycle, 66
- Dienste
  - Definition, 33
  - Hochverfügbarkeit, 48, 81
  - Infrastruktur, 20
    - Siehe* Verteile Infrastrukturdienste
  - Skalierbarkeit, 48, 81
  - Web, 44
- Dienstqualitätsanforderungen
  - Leistung, 46
  - Sicherheit, 47, 48
  - Skalierbarkeit, 47, 48
  - Verfügbarkeit, 46, 48
  - Zweckmäßigkeit, 48
- Dienstqualitätsanforderungen
  - Latente Kapazität, 47
  - Zweckmäßigkeit, 47
- Dienstqualitätskomponenten
  - Beschreibung, 81-85
  - Definition, 33
  - Einführung, 24-26
- Directory Preparation Tool
  - als Dienstqualitätskomponente, 26
  - Beschreibung, 85
- Directory Proxy Server
  - als Dienstqualitätskomponente, 25
  - als Systemkomponente, 48
  - Beschreibung, 83
- Directory Server
  - als Infrastrukturdienst, 40-41
  - als Systemdienstkomponente, 23
  - Beschreibung, 78



## E

- Einführungsszenarien, Java ES
  - Aufrüstung, 30
  - Definition, 32
  - Erweiterung, 30
  - Informationen, 30-32
  - neues System, 30
  - Verbesserung, 30
- Einzelidentität
  - Definition, 62
  - Einführung, 58
- EJB-Komponenten, 44
- Endbenutzer
  - Definition, 33
  - verteilte Anwendungen und, 20
- Entwicklung
  - als Bereitstellungsaufgabe, 70
  - Definition, 73
- Ermitteln installierter Software, 56

## G

- Gemeinsam genutzte Komponenten, 85
  - Definition, 33
  - Einführung, 26
- Geschäftsdienste
  - Definition, 54
  - Präsentationsschicht und, 43
- Glossary, Verweis auf, 15

## H

- Hardware, Java ES-Einführungsszenarien und, 31
- High Availability Session Store
  - als Dienstqualitätskomponente, 25
  - Beschreibung, 82

## I

- ICU (International Components for Unicode), 86
- Identität
  - Dienste, 21, 58-62
  - Einzelbenutzer, 58-59

## Identität (Fortsetzung)

- Verwaltung, 58
- Implementierungsspezifikationen, 68
- Infrastruktur
  - Dienstabhängigkeiten
    - Siehe* Verteilte Dienste
  - für verteilte
    - Unternehmensanwendungen, 20
- Instant Messaging
  - als Infrastrukturdienst, 40-41
  - als Systemdienstkomponente, 23
  - Beschreibung, 78
- Integration
  - Dienste, 39
  - Funktionen
    - Siehe* Integrationsfunktionen
  - Java ES-Einführungsszenarien und, 31
- Integrationsfunktionen
  - Einführung, 21
  - gemeinsam genutzte Komponenten, 21
  - Identität und Sicherheit, 22, 58-62
  - integriertes Installationsprogramm, 22, 55-57

## J

- J2EE
  - Komponenten, 43
  - Plattform, 24
  - verteiltes Komponentenmodell, 44
- J2ME-Plattform, 43
- J2SE (Java 2 Platform, Standard Edition), 26, 86
- JAF (JavaBeans™ Activation Framework), 86
- JATO (Java Studio Enterprise Web Application Framework), 86
- Java Servlet-Komponenten, 43
- JavaHelp, 86
- JavaMail, 86
- JAX-RPC, 86
- JAXB (Java Architecture for XML Binding), 86
- JAXP (Java API for XML Processing), 86
- JAXR (Java API for XML Registries), 86
- JDMK (Java Dynamic Management™ Kit), 86
- JMS (Java Message Service), 23
- JSP-Komponenten, 43
- JSS (Java Security Services), 26, 86

## K

- Kommunikationsdienste, 21
- Komponenten
  - Abhängigkeiten, 41-42
  - Definition, 32
  - Dienstqualität, 24-26, 81-85
  - EJB, 44
  - Ermitteln installierter Versionen von, 56
  - gemeinsam genutzte, 26, 85-86
  - J2EE, 43
  - Servlet, 43
  - System
    - Siehe* Systemkomponenten
  - Systemdienst, 22-24
  - Systemdienste, 76-80
  - und Infrastrukturdienste, 40
  - Verfügbarkeit, 24-25
  - verteilt, 20
  - Verwaltung, 26
  - Zugriff, 25-26
- Komponentent, JSP, 43
- KT Search Engine (KTSE), 86

## L

- Latente Kapazität, Anforderungen, 47
- LDAP, 44, 62
- LDAP C SDK, 86
- LDAP Java SDK, 86
- Lebenszyklusphase
  - Bereitstellungsnachbereitung, 29, 72
  - Bereitstellungsvorbereitung, 29, 65
- Lebenszyklusphase, Bereitstellung, 29
- Leistungsanforderungen, 46
- life-cycle phases, deployment, 66
- Linux, 40-41
- Logische Architekturen
  - Beispiel, 52-53
  - Definition, 54
  - Einführung, 35
  - Infrastrukturdienstebenen und, 37
- Logische Client-Schicht, 43
- Lösungen, Java ES
  - Beispiel, 51
  - benutzerdefiniert und Out-of-the-Box, 51
  - Lebenszyklus, 27-30

## M

- Message Queue
  - als Infrastrukturdienst, 40-41
  - als Systemdienstkomponente, 23
  - Beschreibung, 78
- Messaging-Dienste, 39
- Messaging Server
  - als Infrastrukturdienst, 40-41
  - als Systemdienstkomponente, 23
  - Beschreibung, 79
- Middleware-Dienste, 37
- Migration, Java ES-Einführungsszenarien und, 31

## N

- Netzwerktransportdienste, 38
- NSPR (Netscape Portable Runtime), 26, 86
- NSS (Network Security Services), 26, 86

## P

- Persistenzdienste, 38
- Plattformdienste, 37
- Portal Server
  - als Infrastrukturdienst, 40-41
  - als Systemdienstkomponente, 24
  - Beschreibung, 80
- Portal Server Secure Remote Access
  - als Systemkomponente, 48
  - als Systemqualitätskomponente, 25
  - Beschreibung, 84
- Produktionseinsatz, 71
- Prototyp-Erstellung, 70

## R

- Referenz-Bereitstellungsarchitekturen
  - Definition, 73
  - Einführung, 68
- Richtlinien
  - Autorisierung, 61
  - Definition, 62
- Runtime-Dienste, 39

## S

- SAAJ (SOAP with Attachments API for Java), 86
- SASL (Simple Authentication and Security Layer), 86
- Schichten, logische
  - Anwendungsarchitektur und, 42
  - Client, 43
  - Daten, 44
  - Geschäftsdienst, 44
  - Präsentation, 43
- Schulung, Java ES-Einführungsszenarien und, 31
- Server
  - Definition, 54
  - eigenständige, 44
- Sicherheit
  - Anforderungen, 47, 48
  - Dienste, 21
  - Richtliniendienste, 39
- Single Sign-On
  - Definition, 62
  - Implementierung von, 60-61
  - Infrastrukturdienstebenen und, 39
  - Java ES-Funktion, 23, 59
- Skalierbarkeit
  - Anforderungen, 47, 48
  - Dienste, 48, 81
- SOAP (Apache), 86
- Solaris, 40-41
- solutions, Java ES, architecture, 35
- Sprachunterstützung, 56
- Sun Cluster
  - Agenten, 49
  - als Systemqualitätskomponente, 25
  - als Verfügbarkeitsdienst, 48-49
  - Beschreibung, 81
- Sun Explorer Data Collector, 86
- Sun Java System-Produkte
  - Access Manager
    - Siehe* Access Manager
  - Administration Server
    - Siehe* Administration Server
  - Application Server
    - Siehe* Application Server
  - Calendar Server
    - Siehe* Calendar Server

- Sun Java System-Produkte (Fortsetzung)
  - Communications Express
    - Siehe* Communications Express
  - Connector for Microsoft Outlook
    - Siehe* Connector for 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
    - Siehe* Portal Server
  - Portal Server, Secure Remote Access
    - Siehe* Portal Server, Secure Remote Access
  - Web Server
    - Siehe* Web Server
- Sun Java Web Console, 86
- System
  - Dienste, 19-21
  - Komponenten
    - Siehe* Systemkomponenten
  - Konfiguration, 57
- Systemdienste
  - Definition, 33
  - Informationen, 22
- Systemdienstkomponenten
  - Abhängigkeiten, 41-42
  - Definition, 33
  - Einführung, 22-24
- Systemkomponenten
  - Definition, 33
  - Dienstqualitätskomponenten, 24-26
  - Gemeinsam genutzte Komponenten, 85-86
  - gemeinsam genutzte Komponenten, 26
  - Informationen, 22
  - Systemdienste, 76-80

Systemkomponenten (Fortsetzung)  
Systemdienstkomponenten  
*Siehe* Systemdienstkomponenten

## V

Verbesserungen, *Siehe* Einführungsszenarien

Verfügbarkeit

Anforderungen, 46, 48

Dienste, 48, 81

Verfügbarkeitsdienste, 21

Verfügbarkeitskomponenten

Beschreibung, 81-82

Einführung, 24-25

Verteilt

Anwendungen

*Siehe* Verteilte

Unternehmensanwendungen

verteilt

Dienste

*Siehe* Verteilte Dienste

Verteilte Dienste

Anwendungsebene, 37

Benutzerzusammenarbeit, 39

Identität, 21

Integration, 39

Kommunikation und Zusammenarbeit, 21

Laufzeit, 21

Messaging, 39

Middleware, 37

Netzwerktransport, 38

Persistenz, 38

Plattform, 37, 38

Runtime, 39

Sicherheit, 21, 39

Überblick, 20

Verfügbarkeit, 21

Verwaltung, 21

Web, 21

Zugang, 20

Zugriff, 21

Verteilte Unternehmensanwendungen

Definition, 32

Informationen, 19

Infrastruktur für, 20

Verteilte Dienste, Infrastruktur, 20

Verwaltungsdienste, 21

Verwaltungskomponenten

Beschreibung, 84-85

Einführung, 26

Verzeichnisse

als Benutzerdatenspeicher, 58

Definition, 62

## W

Web Server

als Infrastrukturdienst, 40-41

als Systemdienstkomponente, 24

Beschreibung, 80

Webdienste, 21

Definition, 54

J2EE-Komponenten und, 44

WSCL (Web Services Common Library), 86

## Z

Zugangsdienste, 20

Zugriffsdienste, 21

Zugriffskomponenten

Beschreibung, 82-84

Einführung, 25-26

Zusammenarbeitsdienste, 21

Zweckmäßigkeitserfordernisse, 47, 48